



# UFBA

UNIVERSIDADE FEDERAL DA BAHIA  
ESCOLA POLITÉCNICA  
PROGRAMA DE PÓS GRADUAÇÃO EM  
ENGENHARIA INDUSTRIAL - PEI

DOUTORADO EM ENGENHARIA INDUSTRIAL

JOSÉ CARLOS MARTINS OLIVEIRA

DETECÇÃO E DIAGNÓSTICO DE FALHAS  
EM PROCESSOS DINÂMICOS COM  
REDES NEURAIS SEM PESOS



**SALVADOR**  
**2018**



**WP KXGTU F CF G'HGF GTCN'F C'DCJ K**  
**GUEQNC'RQNK! EP KEC**

**RTQI TCO C'FGR' UI TCFWCYi Q'GO 'GPI GPJ CTK'F WUVTICN**

**LQU 'ECTNQU'O CTVP UQNKXGKTC**

**VGUG'F G FQWQTCFQ**

**FGVGEYi Q'G'FKI P~ UVKQ'F G'HCNJ CU'GO 'RTQEGUQU'F KŽO KQUEQO**  
**TGF GUP GWTCKU'UGO 'RGUQU**

**SALVADOR**  
**2018**

LQU 'ECTNQU'O CTVP UQNK&GKTC

FGVGEYİ Q'GFICI P~ UVKQ'FG'HCNJ CU'GO 'RTQEGUUQU'F PŽO KQEQEQO  
TGF GUP GWTCKU'UGO 'RGUQU

Tese Doutorado apresentada ao Programa de Pós-graduação em Engenharia Industrial, da Universidade Federal da Bahia, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Industrial.

Orientadores: Prof. Marcelo Embiruçu de Souza

Profa. Isabel Sartori

Salvador

2018

Ficha catalográfica elaborada pelo Sistema Universitário de Bibliotecas (SIBI/UFBA), com os dados fornecidos pelo(a) autor(a).

Oliveira, José Carlos Martins

Detecção e Diagnóstico de Falhas com Redes Neurais sem Pesos / José Carlos Martins Oliveira -- Salvador, 2018.

185 f.: il.

Orientadores: Prof. Marcelo Embiruçu de Souza.

Coorientadora: Prof.<sup>a</sup> Isabel Sartori.

Tese (Doutorado – Engenharia Industrial.) Universidade Federal da Bahia. Escola Politécnica, 2018.

1. Detecção e diagnóstico de falhas. 2. Redes neurais sem pesos. 3. Seleção de atributos. 4. Granulação de dados. 5. Sistemas de inferência *fuzzy*. I. Embiruçu, Marcelo. II. Sartori, Isabel. III. Título.

FGVGEYİ Q'G'FKI P~ UVKQ'F'HCNJ CUGO 'RTQEGUUQUF PŽO KEQUEQO  
TGF GUP GWTCKU UGO 'RGUQU

José Carlos Martins Oliveira

TESE SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓSGRADUAÇÃO  
EM ENGENHARIA INDUSTRIAL DA UNIVERSIDADE FEDERAL DA BAHIA COMO  
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
DOUTOR EM CIÊNCIAS EM ENGENHARIA INDUSTRIAL.

Aprovada<sup>1</sup> por:

---

Prof. Cristiano Hora de Oliveira Fontes (UFBA) - Presidente

---

Prof. Robson da Silva Magalhães (UFSB)

---

Prof. Flávio Morais de Assis Silva (UFBA)

---

Profa. Anne Magaly de Paula Canuto (UFRN)

---

Prof. Felipe Maia Galvão França (UFRJ)

---

Prof. Nei Yoshihiro Soma (ITA)

SALVADOR, BA - BRASIL  
ABRIL DE 2018

---

<sup>1</sup> De acordo com o Regimento do Programa, o conjunto de orientadores não participa da banca de avaliação.

*“O futuro pertence àqueles que acreditam na beleza dos seus sonhos”  
Eleanor Roosevelt*

*Dedico este trabalho à minha esposa Adriana, as minhas filhas Nívia, Maria Eduarda e Ana Vitória e aos meus pais José (in memória) e Maria.*

## Agradecimentos

Serei eternamente grato a todos que contribuíram para que eu pudesse vencer esse desafio posto na minha vida. Mas existem aqueles que merecem um agradecimento especial, a saber:

- Ao meu Senhor e Criador, agradecimento eterno pela vida e pelo amor, pois é nele que vivo, respiro e posso cumprir, não só essa, mas todas as missões a mim determinadas;
- Aos meus orientadores, professor Marcelo Embiruçu e professora Isabel Sartori, pela paciência e ensinamentos em todos os momentos solicitados para as devidas orientações. Vocês são bons exemplos de profissionais;
- Aos demais professores do curso que contribuíram para a minha formação pessoal e profissional, em especial, aos Professores Karen Pontes e Cristiano Fontes;
- Aos revisores anônimos dos artigos pela colaboração e sugestões;
- Aos meus amigos e colegas do curso, particularmente, Maísa, Clênia e Agnaldo;
- A Raony Fontes, Iza Silva e Pedro Aragão pelo auxílio técnico na geração da base de dados do CSTR (*Continuous Stirred Tank Reactor*);
- A Tatiane e Robinson, secretários do Colegiado do Programa de Engenharia Industrial (PEI), pelo carinho e atenção nos seus atendimentos;
- Aos meus familiares, em especial, a minha esposa Adriana e as minhas filhas, Nívia, Maria Eduarda e Ana Vitória, que alegram os meus dias e são uma das muitas provas do amor de Deus por mim;
- A Juliana Brito Ribas, testemunha dos meus diários e incansáveis estudos para o desenvolvimento deste trabalho;
- À UESB (Universidade Estadual do Sudoeste da Bahia), à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo apoio financeiro;
- Finalmente, agradeço a Jaime Freire de Souza, egresso do Curso de Ciência da Computação da UESB, pela valiosa colaboração no desenvolvimento dos sistemas de software para os sistemas de FDD propostos, sem você Jaime os desafios seriam bem mais difíceis.



Resumo da Tese apresentada ao PEI/UFBA como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

## DETECÇÃO E DIAGNÓSTICO DE FALHAS EM SISTEMAS DINÂMICOS USANDO REDES NEURAIS SEM PESOS

José Carlos Martins Oliveira

Abril/2018

Orientadores: Marcelo Embiruçu  
Isabel Sartori

Programa: Engenharia Industrial

Sistemas de detecção e diagnóstico de falhas (FDD - *Fault Detection and Diagnosis*) têm sido largamente utilizados em processos industriais. A necessidade de detecção antecipada e segura de falhas em sistemas dinâmicos e reais provocou uma demanda crescente por processos de supervisão que integram os sistemas FDD. Neste contexto, o objetivo principal desta tese é abordar o problema de detecção e diagnóstico de falhas em problemas dinâmicos univariável e multivariáveis com base nas Redes Neural Sem Pesos (RNSP). As RNSP utilizam neurônios baseados em dispositivos de memórias RAM (*Random Access Memories*, memórias de acesso randômico ou aleatório) para o aprendizado das características intrínsecas nos dados de treinamento. Essas redes apresentam algoritmos de aprendizagem rápidos e flexíveis, precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento das redes e um potencial elevado para o reconhecimento e classificação de padrões. Como resultado da pesquisa realizada, são propostos três sistemas de detecção e diagnóstico de falhas baseados no modelo neural sem pesos, conhecido como dispositivo WiSARD (*Wilkie, Stonham e Aleksander's Recognition Device*, dispositivo de reconhecimento de Wilkie, Stonham e Aleksander). Os sistemas propostos contam também com as contribuições da seleção de atributos, de modelos estatísticos básicos e da lógica *fuzzy* para a formação dos padrões comportamentais apresentados às RNSP usadas. Para a validação dos sistemas propostos foram testados três estudos de caso. Um deles se refere a um problema real univariável com dados obtidos pelos sensores de temperatura do compressor de gás para a venda em uma Unidade de Processamento de Gás Natural (UPGN) da Petrobras, localizada no município de Pojuca na Bahia (UPGN-3-Bahia). O segundo simula uma planta industrial considerada como um *benchmark* na área de detecção e diagnóstico de falhas, e conhecida como *Tennessee Eastman Process* (TEP). O terceiro e último estudo de caso simula um reator tanque agitado continuamente (CSTR, *Continuous Stirred Tank Reactor*, reator tanque agitado continuamente). Os resultados apresentados comprovam a boa adaptação das RNSP para o problema de detecção e diagnóstico de falhas, com percentuais de acertos, na classificação, acima de 98%.

**Palavras-chave:** Detecção e Diagnóstico de Falhas; Redes Neurais sem Pesos; Seleção de Atributos; Granulação de Dados; Sistemas de Inferência *Fuzzy*.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

## **FAULT DETECTION AND DIAGNOSIS IN DYNAMIC SYSTEMS USING WEIGHTLESS NEURAL NETWORKS**

José Carlos Martins Oliveira

Abril/2018

Advisors: Marcelo Embiruçu de Souza  
Isabel Sartori

Programme: Industrial Engineering

Fault Detection and Diagnosis (FDD) systems have been widely used in industrial processes. The need of detection anticipated and secure of failure in dynamic and real systems provoke to a growing demand for supervisory processes that integrate FDD systems. In this context, the main objective of this thesis is to approach the problem of detecting and diagnosing failures in univariate and multivariate dynamic problems based on Weightless Neural Networks (WNN). The RNSP uses neurons based on RAM (Random Access Memories) devices to learn the intrinsic characteristics in the training data. These networks use fast and flexible learning algorithms, which provide accurate and consistent results, without the need for residual generation or network retraining, and therefore they have great potential use for pattern recognition and classification. As result of the research, three systems of fault detection and diagnosis based on the weightless neural model, known as WiSARD device (**W**ilkie, **S**tonham e **A**leksander's **R**ecognition **D**evice, dispositivo de reconhecimento de Wilkie, Stonham e Aleksander) are proposed. The proposed systems also count with the contributions of the attribute selection, basic statistical models and fuzzy logic for the formation of behavioral patterns presented to the RNSP used. For the validation of the proposed systems, three case studies were tested. One of them refers to a real univariate problem with data obtained by temperature sensors of the gas of sale compressor of a Petrobras Natural Gas Processing Unit (NGPU) located in the city of Pojuca in Bahia (NGPU-3, Bahia). The second simulates an industrial plant considered as a benchmark in the area of fault detection and diagnosis and known as the Tennessee Eastman Process (TEP). The third and final case study simulates a continuous stirred tank reactor (CSTR, Continuous Stirred Tank Reactor). The results show the good adaptation of the RNSP to the problem of detection and diagnosis of failures with percentage of correctness in the classification above 98%.

**Keywords:** Fault Detection and Diagnosis; Neural Networks Weightless; Selection of Attributes; Granulation of Data; Fuzzy Inference Systems.

# Sumário

Lista de Figuras .....	i
Lista de Tabelas.....	iii
Publicações.....	vi
Registro de Programas de Computador.....	vi
<b>Capítulo 1 : Introdução.....</b>	<b>8</b>
1.1 Caracterização do Problema .....	9
1.2 Originalidade e Contribuições .....	11
1.3 Objetivos.....	12
1.1.1 Objetivo Geral .....	12
1.1.2 Objetivos Específicos .....	12
1.4 Organização do Trabalho .....	13
Abreviaturas .....	13
Referências .....	14
<b>Capítulo 2 : Pressupostos Teóricos e Metodológicos.....</b>	<b>20</b>
2.1 Detecção e Diagnóstico de Falhas em Processos Industriais .....	21
2.1.1 Supervisão e Proteção em Processos Industriais .....	21
2.2 Redes Neurais Artificiais.....	25
2.2.1 Redes Neurais Artificiais com Pesos.....	27
2.2.2 Redes Neurais Artificiais sem Pesos .....	28
2.3 Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos.....	29
2.3.1 Reconhecimento de Padrões.....	29
2.3.2 Detecção e Diagnóstico de Falhas com Redes Neurais sem Pesos .....	30
2.4 Materiais e Métodos .....	32
2.5 Estudos de Casos .....	33
2.5.1 Unidade de Processamento de Gás Natural (UPGN) .....	33
2.5.2 <i>Tennessee Eastman Process</i> (TEP) .....	35
2.5.3 <i>Continuous Stirred Tank Reactor</i> (CSTR) .....	36
2.6 Sistemas Propostos .....	37
Abreviaturas .....	38
Referências .....	39
<b>Capítulo 3 : Detecção e Diagnóstico de Falhas em Sistemas Dinâmicos Usando Redes Neurais Sem Pesos WiSARD.....</b>	<b>45</b>
3.1 Introdução.....	46

3.2	Redes Neurais sem Pesos .....	48
3.2.1	A Rede WiSARD .....	49
3.2.2	A Técnica de <i>Bleaching</i> .....	50
3.3	Sistema de Detecção e Diagnóstico de Falhas FDD-WiSARD Proposto .....	51
3.3.1	Estrutura do Sistema.....	51
3.3.2	Pré-Processamento dos Padrões de Entrada .....	52
3.3.2.1	Seleção de Atributos .....	52
3.3.2.2	Mapeamento dos Padrões de Entrada .....	53
3.3.3	Estrutura da Rede Neural sem Pesos .....	55
3.3.4	Filtro de Agrupamento .....	56
3.3.5	Metodologia para o Treinamento e Teste do Sistema FDD-WiSARD .....	57
3.4	Estudos de Caso.....	58
3.4.1	Unidade de Processamento de Gás Natural (UPGN) .....	58
3.4.1.1	Resultados e Análises - UPGN .....	59
3.4.2	O Processo <i>Tennessee Eastman</i> (TEP).....	66
3.4.2.1	Resultados e Análises - TEP .....	69
3.5	Conclusões Parciais .....	77
	Abreviaturas .....	78
	Notações .....	80
	Símbolos Gregos .....	81
	Referências .....	82
	<b>Capítulo 4 : Detecção e Diagnóstico de Falhas Baseados em Redes Neurais Sem Pesos Granulares e Seleção de Atributos .....</b>	<b>89</b>
4.1	Introdução.....	90
4.2	Redes Neurais Artificiais.....	92
4.2.1	Redes Neurais sem Pesos .....	92
4.2.1.1	Modelo RAM.....	92
4.2.1.2	A Rede WiSARD.....	93
4.2.1.3	Modelo PLN .....	95
4.2.1.4	Modelo MPLN.....	95
4.3	O Sistema de Detecção e Diagnóstico de Falhas FDD-RNSPG Proposto .....	96
4.3.1	Estrutura do Sistema.....	96
4.3.2	Seleção de Atributos.....	97
4.3.3	Granulação <i>Fuzzy</i> dos Atributos Seleccionados.....	98
4.3.4	Estrutura da Rede Neural sem Pesos .....	100

4.3.5	Filtro de Agrupamentos .....	103
4.3.6	Metodologia para o Treinamento e Teste do Sistema FDD-RNSPG .....	104
4.4	Estudos de Caso.....	105
4.4.1	<i>Tennessee Eastman Process (TEP)</i> .....	105
4.4.1.1	Resultados e Análises - TEP.....	106
4.4.2	Reator de Tanque Agitado Continuamente (CSTR).....	112
4.4.2.1	Resultados e Análises - CSTR.....	114
4.5	Conclusões Parciais .....	121
	Abreviaturas .....	122
	Notações .....	123
	Símbolos Gregos .....	125
	Referências .....	126
	<b>Capítulo 5 : Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Probabilísticas.....</b>	<b>131</b>
5.1	Introdução.....	132
5.2	Redes Neurais sem Pesos .....	133
5.2.1	O Modelo RAM.....	134
5.2.2	O Modelo PLN .....	136
5.2.3	O Modelo MPLN.....	137
5.2.4	Modelo pRAM.....	138
5.3	O Sistema Neural sem Pesos Probabilístico FDD-pRAM Proposto .....	140
5.3.1	Estrutura do Sistema.....	140
5.3.2	Seleção de Atributos.....	141
5.3.3	Sistema de Inferência <i>Fuzzy</i> para os Atributos Seleccionados .....	142
5.3.4	Estrutura da Rede Neural sem Pesos .....	143
5.3.5	Filtro de Agrupamentos .....	145
5.3.6	Análise de Desempenho .....	145
5.4	Estudos de Caso.....	145
5.4.1	O Processo Tennessee Eastman (TEP).....	146
5.4.1.1	Resultados e Análises - TEP.....	148
5.4.2	Reator Tanque Agitado Continuamente (CSTR) .....	153
5.4.2.1	Resultados e Análises - CSTR.....	155
5.5	Conclusões Parciais .....	161
	Abreviaturas .....	162
	Notações .....	163

Símbolos Gregos .....	164
Referências .....	165
<b>Capítulo 6 : Conclusões e Sugestões para Trabalhos Futuros .....</b>	<b>169</b>
6.1 Objetivos Alcançados .....	170
6.2 Contribuições.....	170
6.3 Sugestões para Trabalhos Futuros .....	171
Abreviaturas .....	172

## Lista de Figuras

<b>Figura 1.1:</b> Classificação dos Modelos de Sistemas de FDD [D’Angelo (2010)].	10
<b>Figura 2.1:</b> Processo Industrial com FDD Acoplado.	21
<b>Figura 2.2:</b> Conceitos Relacionados à Falhas e Falhas [adaptado de Silva (2008)].	23
<b>Figura 2.3:</b> Evolução Temporal das Falhas – (a) Falhas Abruptas; (b) Falhas Incipientes e (c) Falhas Intermitentes [Adaptado de Andrade (2012)].	24
<b>Figura 2.4:</b> Classificação de Falhas quanto à Localização [Adaptado de Andrade (2012)].	24
<b>Figura 2.5:</b> Um Neurônio com Pesos Típico (Adaptado de Oliveira, 2002).	27
<b>Figura 2.6:</b> Esquema do Processamento do Gás Natural da UPGN em Estudo [Adaptado de Sartori, 2012].	34
<b>Figura 2.7:</b> Diagrama Esquemático do <i>Tennessee Eastman Process</i> (TEP).	35
<b>Figura 2.8:</b> Diagrama Esquemático do Reator CSTR.	36
<b>Figura 2.9:</b> Diagrama Estrutural para os Sistemas FDD Propostos.	37
<b>Figura 3.1:</b> Estrutura da Rede e do Neurônio RAM [Oliveira <i>et al.</i> , 2017].	48
<b>Figura 3.2:</b> Representação Esquemática de um Discriminador.	49
<b>Figura 3.3:</b> WiSARD - Reconhecimento com $k$ Discriminadores [Adaptado de Aleksander, (2009)].	50
<b>Figura 3.4:</b> Estrutura do Sistema Neural sem Pesos: FDD-WiSARD.	52
<b>Figura 3.5:</b> Metodologia para os Treinamento e Testes com o Sistema FDD-WiSARD.	56
<b>Figura 3.6:</b> Metodologia para os Treinamento e Testes com o Sistema FDD-WiSARD.	57
<b>Figura 3.7:</b> Diagrama Estrutural do Compressor de Gás para a Venda na UPGN em Estudo [adaptado de Andrade (2012)].	58
<b>Figura 3.8:</b> Dados Utilizados no Treinamento e Validação dos Sistemas FDD para o caso UPGN. Na Validação: Entradas com Saídas Classificadas com Normal; Entradas com Saídas Classificadas com Desvio; Entradas com Saídas Classificadas com Rompido.	61
<b>Figura 3.9:</b> Diagrama Esquemático do <i>Tennessee Eastman Process</i> (TEP).	67
<b>Figura 3.10:</b> Dados Utilizados no Treinamento dos Sistemas FDD para o CSTR: Modos de Operação “Normal” e “Falha 4”.	71
<b>Figura 4.1:</b> Estrutura de um Sistema Neural Granular [adaptado de Leite <i>et al.</i> (2009)].	90
<b>Figura 4.2:</b> Estrutura da Rede e do Neurônio RAM (Oliveira <i>et al.</i> , 2017).	93
<b>Figura 4.3:</b> Representação Esquemática da Rede WiSARD.	94
<b>Figura 4.4:</b> Estrutura do Sistema Neural sem Pesos: FDD-RNSPG.	97
<b>Figura 4.5:</b> Criação e Adaptação de Grânulos [adaptado de Leite <i>et al.</i> (2009)].	99

<b>Figura 4.6:</b> Arquiteturas da Rede Neural sem Pesos (RNSP).....	102
<b>Figura 4.7:</b> Esquemas de Detecção e Diagnóstico com o Sistema FDD-RNSPG. ....	103
<b>Figura 4.8:</b> Metodologia para os Treinamentos e Testes com o Sistema FDD-RNSPG. ....	104
<b>Figura 4.9:</b> Diagrama Esquemático do <i>Tennessee Eastman Process</i> (TEP) (Oliveira <i>et al.</i> , 2017).....	105
<b>Figura 4.10:</b> Dados Utilizados no Treinamento dos Sistemas FDD para o TEP: Modos de Operação “Normal” e “Falha 4”.....	108
<b>Figura 4.11:</b> Diagrama Esquemático do Reator CSTR.....	112
<b>Figura 4.12:</b> Dados Utilizados no Treinamento dos Sistemas FDD para o CSTR: Modos de Operação “Normal” e “Falha 4”.....	116
<b>Figura 5.1:</b> Estrutura da Rede e do Neurônio RAM (Oliveira <i>et al.</i> , 2017).....	134
<b>Figura 5.2:</b> Representação Esquemática da Rede WiSARD [Adaptado de Oliveira <i>et al.</i> (2017)].....	135
<b>Figura 5.3:</b> Processo para Computar a Saída de um Nodo pRAM com Entradas Binárias e Saída Real [Adaptado de Ludermir <i>et al.</i> (1999)].....	138
<b>Figura 5.4:</b> Processo para Computar a Saída de um Nodo pRAM com Entradas Contínuas e Saída Real [Adaptado de Ludermir <i>et al.</i> (1999)].....	139
<b>Figura 5.5:</b> Estrutura do Sistema Neural sem Pesos: FDD-pRAM.....	141
<b>Figura 5.6:</b> Diagrama Esquemático do <i>Tennessee Eastman Process</i> (Oliveira <i>et al.</i> , 2017). .....	146
<b>Figura 5.7:</b> Dados Utilizados no Treinamento dos Sistemas FDD para o TEP: Modos de Operação “Normal” e “Falha 2”.....	149
<b>Figura 5.8:</b> Diagrama Estrutural do Reator CSTR.....	153
<b>Figura 5.9:</b> Dados Utilizados no Treinamento dos Sistemas FDD para o CSTR: Modos de Operação “Normal” e “Falha 2”.....	156



## Lista de Tabelas

<b>Tabela 3.1:</b> Amostras dos Dados de Treinamento e Validação do Sistema FDD-WiSARD..	60
<b>Tabela 3.2:</b> Resultados da Detecção e do Diagnóstico de Falhas pelo Sistema FDD-WiSARD com Base nos Modelos de Pré-Processamento para as Redes WP e WC .....	61
<b>Tabela 3.3:</b> Parâmetros de Melhor Desempenho do Sistema FDD-WiSARD para cada par RNSP/Modelo de Mapeamento Mostrado da Tabela 3.2.....	62
<b>Tabela 3.4:</b> Detecção e Diagnóstico de Falhas do Sistema FDD-WiSARD com o Modelo de Mapeamento 2 e Rede WiSARD Contador (WC-2) .....	62
<b>Tabela 3.5:</b> Matriz de Probabilidade de Acerto (Confusão) do Diagnóstico do Sistema Obtido com a Rede WiSARD Contador para o Modelo de Mapeamento 2 (WC-2) .....	63
<b>Tabela 3.6:</b> Matriz de Confusão da Detecção do Sistema Obtida com a Rede WiSARD Contador para o Mapeamento do Modelo 2 (CW-2) .....	65
<b>Tabela 3.7:</b> Variáveis Medidas no TEP.....	68
<b>Tabela 3.8:</b> Falhas Simuladas no TEP.....	69
<b>Tabela 3.9:</b> Amostras dos Dados de Treinamento e Validação do Sistema FDD-WiSARD..	70
<b>Tabela 3.10:</b> Percentuais de Acerto das RNSP para o TEP (%) .....	71
<b>Tabela 3.11:</b> Parâmetros de Melhor Desempenho do Sistema FDD-WiSARD para cada par RNSP/Modelo de Mapeamento Mostrado da Tabela 3.10.....	72
<b>Tabela 3.12:</b> Matriz de Classificação (Confusão) para a Rede WiSARD Contador (WC-0) com Variação de +3% na Taxa de Produção: Algoritmo de Seleção de Atributos RecPun Modificado .....	73
<b>Tabela 3.13:</b> Matriz de Classificação (Confusão) para a Rede WiSARD Contador (W-0) com Variação de +3% na Taxa de Produção: Algoritmo de Seleção de Atributos RecPun Original .....	73
<b>Tabela 3.14:</b> Atributos Selecionados por Classe pelos Algoritmos RecPun Original e RecPun Modificado, para WC-0 com Variação de +3% na Taxa de Produção .....	74
<b>Tabela 3.15:</b> Matriz de Confusão da Detecção da Rede Contador WiSARD (CW-0) com variação de + 3% na Taxa de Produção: Algoritmo de Seleção de Atributos RecPun Modificado .....	75
<b>Tabela 3.16:</b> Indicadores de Qualidade na Validação do Sistema FDD-WiSARD (WC-0 com Variação de +3% na Taxa de Produção) e Comparação com Soares e Galvão (2010).....	76
<b>Tabela 4.1:</b> Falhas Simuladas no TEP (Oliveira <i>et al.</i> , 2017).....	106
<b>Tabela 4.2:</b> Percentuais (%) de Acerto das RNSP para a Aplicação TEP .....	108

<b>Tabela 4.3:</b> Melhores Parâmetros Encontrados pelo Sistema FDD-RNSPG com Base no Modelo de Arquitetura 1 com Taxa de Variação na Taxa de Produção para a Aplicação TEP .....	108
<b>Tabela 4.4:</b> TEP: Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e Variação de -3% na Taxa de Produção e Uso do Algoritmo de Seleção de Atributos RecPun Modificado .....	109
<b>Tabela 4.5:</b> TEP: Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e Variação de -3% na Taxa de Produção e Uso do Algoritmo de Seleção de Atributos RecPun Original.....	110
<b>Tabela 4.6:</b> Indicadores de Qualidade na Validação do Sistema FDD-RNSPG com a Arquitetura 1 e Variação de -3% na Taxa de Produção de <i>G</i> e <i>H</i> e Comparação com Oliveira <i>et al.</i> (2017) e Soares e Galvão (2010).....	111
<b>Tabela 4.7:</b> Variáveis Medidas para o CSTR (Singhal e Seborg, 2002).....	113
<b>Tabela 4.8:</b> Modos de Operação para o CSTR (Singhal e Seborg, 2002).....	113
<b>Tabela 4.9:</b> Resultados das RNSP para a Aplicação CSTR .....	117
<b>Tabela 4.10:</b> Melhores Parâmetros Encontrados pelo Sistema FDD-RNSPG com Base na Arquitetura 1 na Aplicação CSTR .....	117
<b>Tabela 4.11:</b> Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e o Algoritmo de Seleção de Atributos RecPun Modificado .....	119
<b>Tabela 4.12:</b> Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e o Algoritmo de Seleção de Atributos RecPun Original .....	120
<b>Tabela 5.1:</b> Modos de Operação para o <i>Tennessee Eastman Process</i> (TEP).....	147
<b>Tabela 5.2:</b> Falhas Simuladas no TEP (Oliveira <i>et al.</i> , 2017).....	147
<b>Tabela 5.3:</b> Percentuais de Acerto para a Rede WiSARD pRAM para a Aplicação TEP (%) .....	150
<b>Tabela 5.4:</b> Parâmetros de Melhor Desempenho para o Sistema FDD-pRAM com Base nos Modelos de Arquitetura e na WiSARD com nodos pRAM para a Aplicação TEP .....	150
<b>Tabela 5.5:</b> Matriz de Classificação para a RNSP WiSARD com nodos pRAM e Variação de -3% na Taxa de Produção com Algoritmo de Seleção de Atributos RecPun Modificado .....	151
<b>Tabela 5.6:</b> Matriz de Classificação para a RNSP WiSARD com nodos pRAM e Variação de -3% na Taxa de Produção com Algoritmo de Seleção de Atributos RecPun Original .....	151

<b>Tabela 5.7:</b> Indicadores de Qualidade na Validação do Sistema FDD-pRAM com a Rede WiSARD com Nodos pRAM e Variação de -3% na Taxa de Produção/Comparação com Soares e Galvão (2010).....	152
<b>Tabela 5.8:</b> Variáveis Medidas para o CSTR (Singhal e Seborg, 2002).....	154
<b>Tabela 5.9:</b> Modos de Operação para o CSTR (Singhal e Seborg, 2002).....	154
<b>Tabela 5.10:</b> Percentuais de Acerto do Sistema FDD-pRAM para a Aplicação CSTR (%).	157
<b>Tabela 5.11:</b> Parâmetros de Melhor para a Configuração do Sistema FDD-pRAM com Base nos Modelos de Arquitetura para a Rede WiSARD pRAM com a Aplicação CSTR.....	157
<b>Tabela 5.12:</b> Matriz de Classificação para a Rede WiSARD com nodo pRAM com o Algoritmo de Seleção de Atributos RecPun Modificado para a Aplicação CSTR.....	159
<b>Tabela 5.13:</b> Matriz de Classificação para a Rede WiSARD com nodo pRAM com o Algoritmo de Seleção de Atributos RecPun Original para a Aplicação CSTR .....	160

## Publicações

- Oliveira, J. C. M., Pontes, K. V., Sartori, I., Embiruçu, M. (2017). Fault Detection and Diagnosis in dynamic systems using Weightless Neural Networks. *Expert Systems with Applications*, 84, 200-219.
- Oliveira, J. C. M., Fontes, C. H. de O., Sartori, I., Embiruçu, M. (2018). Fault Detection and Diagnosis with Granular Weightless Neural Networks and Selection of Attributes. *To be submitted*.
- Oliveira, J. C. M., Fontes, C. H. O., Sartori, I., Embiruçu, M. (2018). Fault Detection and Diagnosis with Probabilistic Weightless Neural Networks. *To be submitted*.

## Registro de Programas de Computador

- Oliveira, J. C. M., Sartori, I., Embiruçu, M., Souza, J. F. de S. (2017). Sistema de Detecção e Diagnóstico de Falhas: FDD - WiSARD. *Diretoria de Patentes, Programas de Computador e Topografias de Circuitos Integrados, Instituto Nacional da Propriedade Industrial*, Brasil. Registro concedido em 01.08.2017.

# CHAPTER I

## Chapter 1: Introduction

### Abstract

The purpose of this chapter is to contextualize the research problem. For this, section 1.1 it is characterized the problem of detection and diagnosis of failures in the context of the supervisory and control systems in industrial processes. This characterization focuses on the motivation, relevance, definitions, methods and approaches used for the development of these systems. Section 1.2 presents originality and research-related contributions. Section 1.3 defines the general and specific objectives to be achieved with the development of the research. Finally, section 1.4 shows the organization of the text as a whole and the approach of the other chapters of this thesis.

**Keywords:** Detection and Diagnosis of Failures; Supervision and Control; Models and Approaches; Weightless Neural Networks.

# CAPÍTULO I

## Capítulo 1: Introdução

### Resumo

Este capítulo tem por finalidade contextualizar o problema da pesquisa realizada. Para isso, na seção 1.1 é caracterizado o problema de detecção e diagnóstico de falhas no contexto dos sistemas de supervisão e controle em processos industriais. Esta caracterização foca na motivação, relevância, definição e nos métodos e abordagens usados para o desenvolvimento dos referidos sistemas. A seção 1.2 apresenta a originalidade e as contribuições relacionadas à pesquisa. A seção 1.3 define o objetivo geral e os específicos alcançados com os trabalhos desenvolvidos. Por último, a seção 1.4 expõe a organização do texto como um todo e demais capítulos que compõe esta tese.

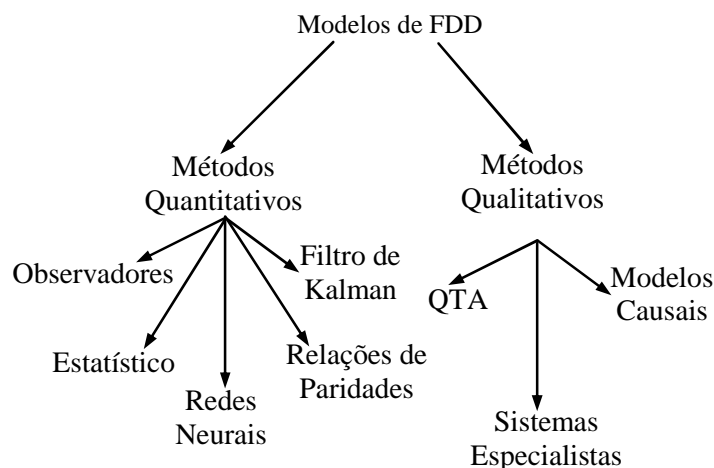
**Palavras-chave:** Detecção e Diagnóstico de Falhas; Supervisão e Controle; Modelos e Abordagens; Redes Neurais sem Pesos.

## 1.1 Caracterização do Problema

Com os avanços tecnológicos, os processos industriais se tornam mais produtivos e confiáveis e com automações mais complexas e com maior disponibilidade de dados. Consequentemente, aumenta a necessidade de sistemas de supervisão e controle mais eficientes, com destaque para a detecção e diagnóstico de falhas (FDD, *Fault Detection and Diagnosis*, detecção e diagnóstico de falhas) em ambientes dinâmicos. Na indústria, as primeiras aparições, com essa denominação datam de 1970 (Korbicz e Witczak, 2005). Nessa época, os sistemas de detecção e diagnóstico de falhas eram muito simples, mas já apresentavam ganhos materiais e humanos no processo de produção (Pietrosanto e Betta, 2000). Atualmente, os modelos computacionais, com destaque para os “sistemas inteligentes”, permitem o desenvolvimento de sistemas de FDD mais completos e seguros. A detecção antecipada e segura de falhas em sistemas dinâmicos e reais diminui os riscos de mau funcionamento ou paradas não programadas do sistema. Consequentemente, aumenta a confiabilidade nos equipamentos, evitam as perdas materiais e, até mesmo, acidentes de trabalho e ambientais (Blázquez e Miguel, 2005; Burns, 2006; Chiang *et al.*, 2001; Fan *et al.*, 2013; Yang e Liu, 1998).

Para Isermann e Ballé (1997), a detecção de falhas é consequência de um comportamento indesejável ou inaceitável do sistema e consiste na presença de uma falha e do tempo de sua percepção. Já o diagnóstico de falhas é a determinação de características tais como: tipo; localização; tamanho; causa; e comportamentos temporal das falhas detectadas. Essas características variam de acordo com o objetivo de cada sistema, sendo que a localização e a causa são as mais abordadas na literatura (Isermann e Ballé, 1997; Venkatasubramanian *et al.*, 2003). Também, como etapa complementar, a correção de falhas refere-se à tomada automática ou não de ações apropriadas para restabelecer o funcionamento esperado do sistema ou evitar possíveis perdas materiais e ambientais. Por entender que a correção depende em muito de fatores externos, alheios aos pesquisadores, a maioria dos estudos voltam seus objetivos somente para as etapas de detecção e diagnóstico de falhas.

Os métodos usados para o desenvolvimento de sistemas de detecção e diagnóstico de falhas vão desde o simples acompanhamento de alguma variável do sistema até a análise residual entre os valores real e estimado de uma variável. Essa estimação pode ser obtida por modelos “fenomenológicos” ou “empíricos”. Os modelos empíricos utilizam o conhecimento pré-existente nos dados históricos do processo para obter o valor estimado da variável e são baseados, principalmente, nos métodos computacionais ditos “inteligentes”. Já nos sistemas baseados em modelos fenomenológicos, o comportamento real monitorado é comparado com a variável estimada por um modelo representativo do processo. Nessa comparação supervisionada é obtido um vetor de resíduos usado para detectar a presença de falhas (Sartori, 2012). Num nível maior de detalhamento, os modelos fenomenológicos ou empíricos se subdividem em métodos “quantitativos” e “qualitativos” (Venkatasubramanian *et al.*, 2003a, b), conforme mostra a Figura 1.1.



**Figura 1.1:** Classificação dos Modelos de Sistemas de FDD [D' Angelo (2010)].

Entre os modelos quantitativos destacam-se alguns trabalhos encontrados, tais como: classificadores estatísticos (Luo *et al.*, 2011; Ma *et al.*, 2010; Soares e Galvão, 2010); redes neurais (Lau *et al.*, 2013; Leite *et al.*, 2012; Sartori, 2012; Zarei, 2012); lógica *fuzzy* (Andrade *et al.*, 2011; Andrade, 2012; Lau *et al.*, 2013; Li *et al.*, 2013; Leite *et al.*, 2012; Silva *et al.*, 2012); análise de componentes principais (PCA, *Principal Components Analysis*, análise de componentes principais) (Barragan *et al.*, 2016; Jiang *et al.*, 2013; Lau *et al.*, 2013); método dos mínimos quadrados parciais (Zhang *et al.*, 2010); wavelet transforms (Barragan *et al.*, 2016); observadores de estados e de saídas (Chetouani, 2008; Kalman, 1960); equações e espaço de paridade (Beckerle *et al.*, 2012; Blesa, *et al.*, 2014; Zakharov *et al.*, 2013; Zhong *et al.*, 2015); filtro de Kalman estendido (Kalman, 1960; Patwardhan e Shah, 2006); máquina de vetores de suporte (Deng *et al.*, 2011; Duan *et al.*, 2016; Park *et al.*, 2011; Zhang *et al.*, 2012); e os métodos de identificação e estimação de parâmetros (Johansson *et al.*, 2006; Pouliezos *et al.*, 1989).

No contexto dos modelos qualitativos é possível também destacar muitos trabalhos, entre os quais: sistemas especialistas (Saravanan *et al.*, 2009; Zadeh, 2008; Wang *et al.*, 2012); análise qualitativa de tendências (Maurya, *et al.*, 2007); árvores de falha (Simões Filho, 2006; Nguyen e Lee, 2008); simulação qualitativa (Berleant, 1991); teoria qualitativa de processo (Venkatasubramanian *et al.*, 2003b); e grafos direcionados com sinais, a exemplo das redes Bayesianas (Luo *et al.*, 2012; Ji *et al.*, 2015; Xiao *et al.*, 2014).

Embora existam diversas linhas de pesquisa relacionadas, há um diferencial positivo e crescente das técnicas de base estatística e da inteligência artificial para a solução de problemas de FDD, destacando os sistemas híbridos envolvendo lógica *fuzzy*, algoritmos genéticos e principalmente as redes neurais (Herzog *et al.*, 2009; Hu e Yi, 2016; Liao *et al.*, 2012; Niaki e Abbasi, 2005; Rajpal *et al.*, 2006; Santosh *et al.*, 2007; Sartori *et al.*, 2012; Sharma *et al.*, 2015). Essas técnicas se adéquam bem em sistemas não lineares, por não requerem modelos matemáticos fenomenológicos (Angelov e Yager, 2012; Barty *et al.*, 2006; Bocaniala e da Costa, 2006; Li *et al.*, 2013; Lo *et al.*, 2009; Ma *et al.*, 2010; Rigatos e Zhang, 2009).

De acordo com Sartori *et al.* (2016), a maioria das aplicações de sistemas de FDD está voltada para unidades de geração e distribuição de energia, equipamentos na indústria de processos (reatores, colunas, sensores e atuadores), motores e rolamentos. Dentre as técnicas mais usadas estão: redes neurais, lógica *fuzzy*, PCA, filtro de Kalman, máquina de vetores de suporte, algoritmo genético e sistemas especialistas.



Na sua grande maioria, as abordagens citadas na seção anterior se adaptam bem à detecção e ao diagnóstico de falhas abruptas. No contexto de falhas incipientes em sistemas dinâmicos, a solução do problema de FDD é complexa e os métodos encontrados na literatura referida ainda deixam lacunas, dentre as quais se destacam: problemas multivariáveis e com diversidade no número de classes consideradas; processos não lineares com falhas incipientes presentes em duas ou mais das classes consideradas; e dificuldade na obtenção de dados históricos com aplicações reais. Essas lacunas serviram de motivação para o desenvolvimento deste trabalho, cuja originalidade e contribuições são apresentadas a seguir.

## 1.2 Originalidade e Contribuições

Como pode ser observado na seção anterior, existem diversas contribuições relevantes para o problema de FDD, com destaque para as técnicas de base estatística e dos sistemas inteligentes. Porém, nenhum trabalho encontrado fez uso das “Redes Neurais Sem Pesos” (RNSP) para o problema de detecção e diagnóstico de falhas, principalmente em sistemas dinâmicos multivariáveis. As contribuições com uso de RNSP encontradas na literatura estão voltadas para o problema “clássico” de reconhecimento e classificação de padrões, com ênfase na classificação de imagens digitais. Assim, como ponto de originalidade e contribuição principal deste trabalho, destaca-se o uso das “RNSP” para a proposição de sistemas de FDD em ambientes dinâmicos com aplicações univariável e multivariáveis. Para isso, foram propostos três sistemas de detecção e diagnóstico de falhas incipientes em sistema dinâmicos, sem a necessidade de modelo matemático ou cálculo de resíduos e com uma baixa taxa de alarmes falsos. No entanto, a construção desses sistemas só foi possível mediante contribuições de outras técnicas clássicas para o pré-processamento das variáveis existentes nos padrões de entradas, tais como: algoritmos de seleção de atributos por classe, de modo a diferenciar a importância das variáveis que contribuem para cada classe considerada em cada aplicação (Vale *et al.*, 2010); algoritmos baseados na lógica *fuzzy* (Leite *et al.*, 2009); recorrência de dados (Rosa *et al.*, 2013) e modelos estatísticos para a granulação dos atributos selecionados (Oliveira *et al.*, 2017); e, por último, o uso de filtro de agrupamentos para obtenção de diagnósticos por grupos de falhas com características semelhantes.

As redes neurais sem pesos, propostas inicialmente por Aleksander (1967), são modelos digitais baseados em dispositivos de memória de acesso aleatório (RAM, *Random Access Memories*, memórias de acesso randômico ou aleatório). Diferente dos modelos neurais convencionais, o aprendizado acontece em “memórias” inseridas no próprio neurônio, em forma de tabelas-verdade. A diversidade de uso dessas memórias traz algumas vantagens às RNSP em relação aos modelos com pesos, tais como: semelhança com os sistemas digitais convencionais; algoritmos de aprendizagem rápidos e flexíveis; precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento das redes; capacidade natural de generalização (Clarkson *et al.*, 1992); habilidades de aproximadores universais de funções contínuas (Gorse *et al.*, 1997); e principalmente o grande potencial das RNSP para o reconhecimento e a classificação de padrões (Ludermir *et al.*, 1999).

Para a validação dos sistemas propostos foram testados três estudos de caso. Um deles refere-se a um problema real univariável com dados obtidos pelos sensores de temperatura do compressor de gás para a venda instalado em uma Unidade de Processamento de Gás Natural (UPGN) da Petrobras localizada no município de Pojuca-Bahia (UPGN-3-Bahia). O segundo simula uma planta industrial considerada como um *benchmark* na área de detecção e diagnóstico de falhas e conhecida como *Tennessee Eastman Process* (TEP) (Downs e Vogel, 1993). O terceiro e último estudo de caso simula um reator tanque agitado continuamente (CSTR, *Continuous Stirred Tank Reactor*) não isotérmico (Singhal e Seborg, 2002).

## 1.3 Objetivos

### 1.1.1 Objetivo Geral

Conforme já mencionado anteriormente, o “objetivo principal” de um sistema FDD é o reconhecimento do comportamento anormal (falhas) dos componentes de um processo, ou do próprio processo, através do monitoramento de suas variáveis. No entanto, a maioria dos trabalhos realiza também um diagnóstico das falhas detectadas, que pode resultar na classificação ou em outros fatores mais complexos, conforme interesse do estudo realizado. Neste contexto, o objetivo principal deste trabalho é abordar o problema de detecção e diagnóstico de falhas incipientes em sistemas dinâmicos multivariáveis com uso das redes neurais sem pesos. As arquiteturas das redes neurais sem pesos usadas permitem realizar a detecção e diagnóstico/classificação das falhas de forma simultânea ou o diagnóstico em etapa posterior à detecção, como na maioria dos trabalhos encontrados na literatura relacionada aos sistemas de FDD.

### 1.1.2 Objetivos Específicos

Para uma investigação maior da adaptabilidade dos modelos neurais sem pesos para o problema de FDD em sistemas dinâmicos multivariáveis, foram propostos três sistemas de detecção e diagnóstico de falhas. Esses sistemas são apresentados, respectivamente, com detalhes nos capítulos 3, 4 e 5 e guiados pelos seguintes “objetivos específicos”:

- Revisar os fundamentos teóricos e práticos relacionados aos problemas de detecção, diagnóstico e correção de falhas, com ênfase nas aplicações industriais;
- Revisar os modelos computacionais, com foco nas redes neurais sem pesos;
- Propor sistemas de detecção e diagnóstico de falhas com base nas redes neurais sem pesos e nos dados históricos da aplicação;
- Compreender com detalhes os três estudos de casos considerados para a validação dos sistemas de FDD propostos;
- Formalizar as estruturas e funcionalidades dos sistemas de FDD propostos;
- Desenvolver, em ambiente computacional, os sistemas propostos;
- Estruturar as bases de dados para o treinamento e validação dos sistemas propostos, com base nos estudos de casos considerados;
- Realizar os experimentos e ajustes necessários nos sistemas desenvolvidos com base nos dados históricos das aplicações;
- Avaliar de forma comparativa e reflexiva os resultados obtidos com os existentes na literatura, de forma a perceber o diferencial positivo das detecções e diagnósticos realizados pelos sistemas propostos;
- Propor, a partir dos sistemas desenvolvidos e dos treinamentos realizados, uma arquitetura sistêmica que melhor se adeque para a detecção e o diagnóstico de falhas apresentadas para cada estudo de caso usado.

## 1.4 Organização do Trabalho

Além deste capítulo de introdução, esta tese contém mais 5 capítulos, os quais estão escritos da forma mais discursiva possível para um texto técnico. Os fundamentos teóricos e metodológicos apresentados no capítulo 2 foram complementados e apresentados com mais detalhes nos demais capítulos da tese, conforme conteúdo e sequência de apresentação. Cada capítulo da tese contém seu próprio resumo, objetivos específicos, conclusões parciais, abreviaturas, notações e referências bibliográficas. Em todo o texto, os destaques estão “entre aspas” e as palavras de língua inglesa de difícil tradução ou de uso consagrado foram mantidas na sua forma original em tipo itálico e colocadas, na primeira aparição, o seu significado em língua portuguesa. Assim, os demais conteúdos desta tese estão distribuídos em capítulos, como segue.

O “capítulo 2” apresenta os pressupostos teóricos e metodológicos para a construção dos sistemas de FDD propostos neste trabalho. Como pressupostos teóricos são revisados os modelos usados para detecção e diagnóstico de falhas, com ênfase nos processos industriais e nas redes neurais, com foco nos modelos sem pesos. Também, diante da ausência de trabalhos sobre detecção e diagnóstico de falhas com uso das redes neurais sem pesos, foi apresentada uma revisão da literatura sobre o “reconhecimento” e a “classificação” de padrões com esses modelos neurais sem pesos.

O “capítulo 3” apresenta de forma detalhada o primeiro sistema de FDD, aqui proposto, denominado de FDD-WiSARD (*Detecção e Diagnóstico de Falhas com Base no Dispositivo WiSARD*) e intitulado de “Detecção e Diagnóstico de Falhas em Sistemas Dinâmicos Usando Redes Neurais Sem Pesos” (Oliveira *et al.*, 2017).

O “capítulo 4” apresenta de modo detalhada o segundo sistema de FDD, aqui proposto, denominado de FDD-RNSPG (*Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Granulares*) e intitulado de “Detecção e Diagnóstico de Falhas Baseados em Redes Neurais Sem Pesos Granulares e Seleção de Atributos” (Oliveira *et al.*, 2018a).

O “capítulo 5” apresenta detalhadamente o terceiro sistema de FDD, aqui proposto, denominado de FDD-pRAM (*Detecção e Diagnóstico de Falhas com RAM Probabilísticas*) e intitulado de “Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Probabilísticas” (Oliveira *et al.*, 2018b).

No “capítulo 6” são postas as conclusões sobre os resultados obtidos com a pesquisa realizada, com foco nos sistemas de FDD propostos neste trabalho de tese. Por último, são apresentadas sugestões para trabalhos futuros.

### Abreviaturas

FDD:	<i>Fault Detection and Diagnosis.</i>
Petrobras:	Petróleo Brasileiro S.A.
pRAM:	<i>probabilistic RAM.</i>
RNSP:	Rede Neural Sem Pesos.
UFBA:	Universidade Federal da Bahia.
UPGN:	Unidade de Processamento de Gás Natural.
VG RAM:	<i>Virtual Generalizing RAM.</i>
WiSARD:	<i>Wilkes-Stonham-Aleksander Recognition Device.</i>

PCA:	<i>Principal Components Analysis.</i>
RAM:	<i>Random Access Memory.</i>
CSTR:	<i>Continuous Stirred Tank Reactor.</i>
FDD- WiSARD:	Detecção e Diagnóstico de Falhas com Base no Dispositivo WiSARD.
FDD- RNSPG:	Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Granulares.
FDD- pRAM:	Detecção e Diagnóstico de Falhas com RAM Probabilísticas.

## Referências

- Aleksander, I. (1967). Adaptive systems of logic networks and binary memories. *Proceedings of the Spring Joint Computer Conference*, 30, 707-712.
- Andrade, V. E. (2012). Sistema de detecção e diagnóstico de falhas em sensores de um compressor de gás natural utilizando lógica fuzzy tipo-2. *Dissertação de Mestrado*. Programa de Engenharia Industrial. Universidade Federal da Bahia-UFBA, Salvador, Bahia, Brasil.
- Andrade, V. E., Fontes, C. H., Embiruçu, M. (2011). An interval type-2 fuzzy logic approach for instrument fault detection and diagnosis. *Proceedings of the World Congress on Engineering*, 2, 1008-1012.
- Angelov, P., Yager, R. (2012). A new type of simplified fuzzy rule-based system. *International Journal of General Systems*, 41(2), 163-185.
- Barragan, J. F., Fontes, C. H., Embiruçu, M. (2016). A wavelet-based clustering of multivariate time series using a Multiscale SPCA approach. *Computers & Industrial Engineering*, 95, 144-155.
- Barty, M., Patton, R., Syfert, M., de las Heras, S., Quevedo, J. (2006). Introduction to the DAMADICS actuator FDI benchmark study. *Control Engineering Practice*, 14(6), 577-596.
- Beckerle, P., Schaede, H., Butzek, N., Rinderknecht, S. (2012). Balancing filters: an approach to improve model-based fault diagnosis based on parity equations. *Mechanical Systems and Signal Processing*, 29, 137-147.
- Berleant, D. (1991). Combining qualitative and quantitative simulation: in brief. *Proceedings of the Second Annual Conference on AI, Simulation and Planning in High Autonomy Systems. Integrating Qualitative and Quantitative System Knowledge*, IEEE, 233-240.
- Betta, G., e Pietrosanto, A. (1998). Instrument fault detection and isolation: State of the art and new research trends. *In Instrumentation and Measurement Technology Conference, IEEE*, 1, 483-489.
- Blázquez, L. F., Miguel, L. J. (2005). Additive fault detection in nonlinear dynamic systems with saturation. *ISA Transactions*, 44(4), 515-538.
- Blesa, J., Jiménez, P., Rotondo, D., Nejjari, F., Puig, V. (2014). Fault diagnosis of a wind farm using interval parity equations. *IFAC Proceedings Volumes*, 47(3), 4322-4327.

- Bocaniala, C. D., da Costa, J. S. (2006). Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem. *Control Engineering Practice*, 14(6), 653-669.
- Burns, C. M. (2006). Towards proactive monitoring in the petrochemical industry. *Safety Science*, 44(1), 27-36.
- Chetouani, Y. (2008). Design of a multi-model observer-based estimator for fault detection and isolation (FDI) strategy: application to a chemical reactor. *Brazilian Journal of Chemical Engineering*, 25, 777-788.
- Chiang, L. H.; Russel, E. L.; Braatz, R. D. (2001). Fault detection and diagnosis in industrial systems. 1<sup>a</sup> Ed. London, UK: Springer Vela.
- Clarkson, T. G., Ng., C. K., Gorse, D., and Taylor, J. G. (1992). Learning probabilistic RAM nets using VLSI structures. *IEEE Transactions on Computers* 41(12), 1552-1561.
- D'Angelo, M. F. S. V. (2010). Uma nova formulação fuzzy/bayesiana para detecção de ponto de mudança em séries temporais: aplicações na detecção de falhas incipientes. *Tese de Doutorado*, Programa de Pós-Graduação em Engenharia Elétrica. Universidade Federal de Minas Gerais, Minas Gerais, Brasil.
- Deng, S., Lin, S.-Y., Chang, W.-L. (2011). Application of multiclass support vector machines for fault diagnosis of field air defense gun. *Expert Systems with Applications*, 38(5), 6007-6013.
- Downs, J. J., Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3), 245-255.
- Duan, L., Xie, M., Bai, T., Wang, J. (2016). A new support vector data description method for machinery fault diagnosis with unbalanced datasets. *Expert Systems with Applications*, 64, 239-246.
- Fan, C. F., Yih, S., Tseng, W. H., Chen, W. C. (2013). Empirical analysis of software-induced failure events in the nuclear industry. *Safety Science*, 57, 118-128.
- Gorse, D., Romano-Critchley, D. A., Taylor, J.G. (1997). A pulse-based reinforcement algorithm for learning continuous functions. *Neurocomputing*, 14(4), 319-344.
- Herzog, M. A., Marwala, T., Heyns, P. S. (2009). Machine and component residual life estimation through the application of neural networks. *Reliability Engineering System Safety*, 94(2), 479-489.
- Hu, J., Yi, Y. (2016). A two-level intelligent alarm management framework for process safety. *Safety Science*, 82, 432-444.
- Isermann, R., Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5), 709-719.
- Ji, Z., Xia, Q., Meng, G. (2015). A review of parameter learning methods in bayesian network. In advanced intelligent computing theories and applications. *Springer International Publishing of the series Lecture Notes in Computer Science*, 9227, 3-12.
- Jiang, Q., Yan, X., Zhao, W. (2013). Fault detection and diagnosis in chemical processes using sensitive principal component analysis. *Industrial Engineering Chemistry Research*, 52(4), 1635-1644.
- Johansson, A., Bask, M., Norlander, T. (2006). Dynamic threshold generators for robust fault detection in linear systems with parameter uncertainty. *Automatica*, 42(7), 1095-1106.

- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(1), 35-45.
- Korbicz, J., Witczak, M. (2005). Soft computing in fault detection and isolation. *In EUSFLAT Conference*, 2-3.
- Lau, C. K., Ghosh, K., Hussain, M. A., Hassan, C. C. (2013). Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS. *Chemometrics and Intelligent Laboratory Systems*, 120, 1-14.
- Leite, D., Costa, P., Gomide, F. (2012). Evolving granular neural network for fuzzy time series forecasting. *The 2012 International Joint Conference on Neural Networks*, IEEE, 1-8.
- Li, H., Yu, J., Hilton, C., Liu, H. (2013). Adaptive sliding-mode control for nonlinear active suspension vehicle systems using T-S fuzzy approach. *IEEE Transactions on Industrial Electronics*, 60(8), 3328-3338.
- Liao, Z., Wang, B., Xia, X., Hannam, P. M. (2012). Environmental emergency decision support system based on artificial neural network. *Safety Science*, 50(1), 150-163.
- Ludermir, T. B., de Carvalho, A., Braga, A. P., De Souto, M. C. P. (1999). Weightless Neural Models: A Review de Current and Past Work. *Neural Computing Surveys*, 2, 41-61.
- Luo, H., Wang, Y., Cui, J. (2011). A SVDD approach of fuzzy classification for analog circuit fault diagnosis with FWT as preprocessor. *Expert Systems with Applications*, 38(8), 10554-10561.
- Ma, M. D., Wong, D. S. H., Jang, S. S., Tseng, S. T. (2010). Fault detection based on statistical multivariate analysis and microarray visualization. *IEEE Transactions on Industrial Informatics*, 6(1), 18-24.
- Maurya, M. R., Rengaswamy, R., Venkatasubramanian, V. (2007). Fault diagnosis using dynamic trend analysis: a review and recent developments. *Engineering Applications of Artificial Intelligence*, 20(2), 133-146.
- Nguyen, N. T., Lee, H. H. (2008). Decision tree with optimal feature selection for bearing fault detection. *Journal of Power Electronics*, 8(1), 101-107.
- Niaki, S. T. A., Abbasi, B. (2005). Fault diagnosis in multivariate control charts using artificial neural networks. *Quality and Reliability Engineering International*, 21(8), 825-840.
- Oliveira, J. C. M., Fontes, C. H. de O., Sartori, I., Embiruçu, M. (2018a). Fault Detection and Diagnosis with Granular Weightless Neural Networks and Selection of Attributes. *To be submitted*.
- Oliveira, J. C. M., Fontes, C. H. O., Sartori, I., Embiruçu, M. (2018b). Fault Detection and Diagnosis with Probabilistic Weightless Neural Networks. *To be submitted*.
- Oliveira, J. C. M., Pontes, K. V., Sartori, I., Embiruçu, M. (2017). Fault detection and diagnosis in dynamic systems using weightless neural networks. *Expert Systems with Applications*, 84, 200-219.
- Park, J., Kwon, I.-H., Kim, S.-S., Baek, J.-G. (2011). Spline regression based feature extraction for semiconductor process fault detection using support vector machine. *Expert Systems with Applications*, 38(5), 5711-5718.
- Patwardhan, S. C., Shah, S. L. (2006). From data to diagnosis and control using generalized orthonormal basis filters. *Journal of Process Control*, 16, 157-175.

- Pietrosanto, A., Betta, G. (2000). Instrument Fault Detection and Isolation. State of the Art and New Research Trends. *IEEE Transactions on Instrumentation Measurement*, 49(1), 100-107.
- Pouliezios, A., Stavrakakis, G., Lefas, C. (1989). Fault detection using parameter estimation. *Quality and Reliability Engineering International*, 5(4), 283-290.
- Rajpal, P. S., Shishodia, K. S., Sekhon, G. S. (2006). An artificial neural network for modeling reliability, availability and maintainability of a repairable system. *Reliability Engineering System Safety*, 91(7), 809-819.
- Ricker, N. L. (1995). Optimal steady-state operation of the Tennessee Eastman challenge process. *Computers and Chemical Engineering*, 19(9), 949-959.
- Rigatos, G., Zhang, Q. (2009). Fuzzy model validation using the local statistical approach. *Fuzzy Sets and Systems*, 160(7), 882-904.
- Rosa, R., Gomide, F., Ballini, R. (2013). Evolving hybrid neural fuzzy network for system modeling and time series forecasting. *12th International Conference in Machine Learning and Applications (ICMLA)*, 2, 378-383.
- Santosh, T. V., Vinod, G., Saraf, R. K., Ghosh, A. K., Kushwaha, H. S. (2007). Application of artificial neural networks to nuclear power plant transient diagnosis. *Reliability Engineering System Safety*, 92(10), 1468-1472.
- Saravanan, N., Cholairajan, S., Ramachandran, K. I. (2009). Vibration-based fault diagnosis of spur bevel gear box using fuzzy technique. *Expert Systems with Applications*, 36(2), 3119-3135.
- Sartori, I. (2012). Uma abordagem de desenvolvimento de sistemas de detecção e diagnóstico de falhas baseada em estudos prévios de confiabilidade e sua aplicação ao processamento industrial de gás natural. *Tese de Doutorado*. Programa de Engenharia Industrial. Universidade Federal da Bahia-UFBA, Salvador, Bahia, Brasil.
- Sartori, I., Amaro, C. A., Arduini, F. Z., Souza Júnior, M. B., Embiruçu, M. (2016). Fault detection, diagnosis and correction in industrial processes: a survey of Brazilian research in the global context. *Ciência & Engenharia*, in press.
- Sharma, R. S., Dewan, L., Chatterji, S. (2015). Fault diagnosis methods in dynamic systems: review. *International Journal of Electronics and Electrical Engineering*, 3(6), 465-471.
- Silva, G. C., Palhares, R. M., Caminhas, W. M. (2012). Immune inspired fault detection and diagnosis: A fuzzy-based approach of the negative selection algorithm and participatory clustering. *Expert Systems with Applications*, 39(16), 12474-12486.
- Simões Filho, S. (2006). Análise de árvore de falhas considerando incertezas na definição dos eventos básicos. *Tese de Doutorado*, Programa de Engenharia Civil. Universidade Federal do Rio de Janeiro, Rio de Janeiro RJ, Brasil.
- Singhal, A., Seborg, D. E. (2002). Pattern matching in multivariate time series databases using a moving-window approach. *Industrial & Engineering Chemistry Research*, 41(16), 3822-3838.
- Soares, A. S., Galvão, R. K. H. (2010). Fault detection using linear discriminant analysis with selection of process variables and time lags. *International Conference on Industrial Technology, ICIT2010, IEEE*, 217-222.

- Vale, K. M. O., Neto, A. F., Canuto, A. M. P. (2010). Using a reinforcement-based feature selection method in classifier ensemble. *Proceedings of the 10th International Conference on Hybrid Intelligent Systems, HIS2010*, IEEE, 213-218.
- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N. (2003a). A review of process fault detection and diagnosis: Part II: qualitative models and search strategies. *Computers and Chemical Engineering*, 27(3), 313-326.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S. N. (2003b). A review of process fault detection and diagnosis: Part I: quantitative model-based methods. *Computers and Chemical Engineering*, 27(3), 293-311.
- Wang, W., Li, D. Z., Vrbanek, J. (2012). An evolving neuro-fuzzy technique for system state forecasting. *Neurocomputing*, 87, 111-119.
- Xiao, F., Zhao, Y., Wen, J., Wang, S. (2014). Bayesian network based FDD strategy for variable air volume terminals. *Automation in Construction*, 41, 106-118.
- Yang, S. K., Liu, T. S. (1998). A Petri net approach to early failure detection and isolation for preventive maintenance. *Quality and Reliability Engineering International*, 14(5), 319-330.
- Zadeh, L. A. (2008). Is there a need for fuzzy logic? *Information Sciences*, 178(13), 2751-2779.
- Zakharov, A., Tikkala, V.-M., Jämsä-Jounela, S.-L. (2013). Fault detection and diagnosis approach based on nonlinear parity equations and its application to leakages and blockages in the drying section of a board machine. *Journal of Process Control*, 23(9), 1380-1393.
- Zarei, J. (2012). Induction motors bearing fault detection using pattern recognition techniques. *Expert Systems with Applications*, 39(1), 68-73.
- Zhang, X., Zhou, J., Guo, J., Zou, Q., Huang, Z. (2012). Vibrant fault diagnosis for hydroelectric generator units with a new combination of rough sets and support vector machine. *Expert Systems with Applications*, 39(3), 2621-2628.
- Zhang, Y., Zhou, H., Qin, S. J., Chai, T. (2010). Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. *IEEE Transactions on Industrial Informatics*, 6(1), 3-10.
- Zhong, M., Song, Y., Ding, S. X. (2015). Parity space-based fault detection for linear discrete time-varying systems with unknown input. *Automatica*, 59, 120-126.



## CHAPTER II

### Chapter 2: Theoretical and Methodological Assumptions

#### Abstract

The objective of this chapter is to present the theoretical and methodological assumptions necessary for the construction of the fault detection and diagnosis systems proposed in this thesis. For this, section 2.1 presents the main concepts and definitions about fault detection, diagnosis and correction, with emphasis on industrial production systems. In section 2.2 the artificial neural network models are reviewed, with focus on the weightless neural network models. In section 2.3 it is presented the state of the art about fault detection and diagnosis with weightless neural networks with focus on the “recognition” and “classification” of patterns. Section 2.4 presents the materials and methods used in the development of this work. Section 2.5 summarizes the three case studies. Finally, section 2.6 presents the general structure of the proposed FDD systems, which are detailed in chapters 3, 4 and 5 of this thesis.

**Keywords:** Fault Detection and Diagnosis; Supervisory and Protection Systems; Weightless Neural Networks; Pattern Recognition.

## CAPÍTULO II

### Capítulo 2: Pressupostos Teóricos e Metodológicos

#### Resumo

O objetivo deste capítulo é apresentar os pressupostos teóricos e metodológicos necessários para a construção dos sistemas de detecção e diagnóstico de falhas propostos nesta tese. Para isso, na seção 2.1 são apresentados os principais conceitos e definições sobre detecção, diagnóstico e correção de falhas, com destaque para os sistemas de produção industrial. Na seção 2.2 são revisados os modelos de redes neurais artificiais, com foco nos modelos neurais sem pesos. Na seção 2.3 é apresentado o estado da arte sobre detecção e o diagnóstico de falhas com redes neurais sem pesos com foco no “reconhecimento” e na “classificação” de padrões. Na seção 2.4 são apresentados os materiais e métodos usados no desenvolvimento deste trabalho. Na seção 2.5 é feita uma apresentação resumida dos três estudos de casos. Por último, a seção 2.6 apresenta a estrutura geral dos sistemas de FDD propostos, os quais são detalhados nos capítulos 3, 4 e 5 desta tese.

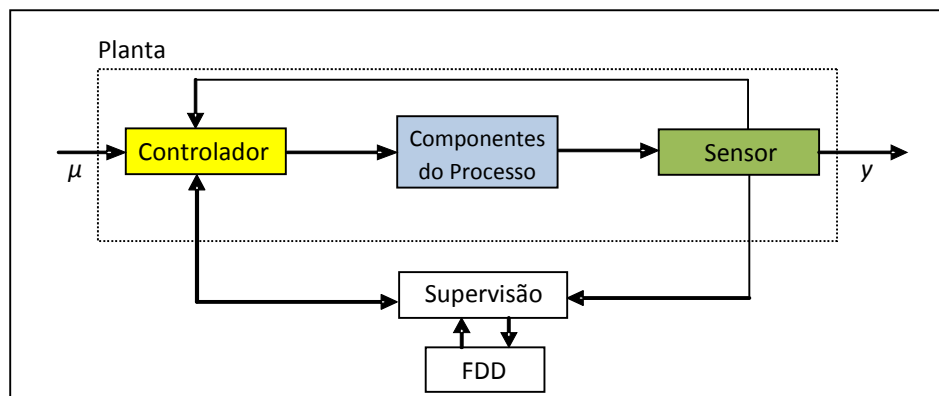
**Palavras-Chave:** Detecção e Diagnóstico de Falhas; Sistemas de Supervisão e Proteção; Redes Neurais sem Pesos; Reconhecimento de Padrões.

## 2.1 Detecção e Diagnóstico de Falhas em Processos Industriais

Historicamente, os primeiros sistemas de detecção, diagnóstico e correção de falhas (FDDC, *Fault Detection, Diagnosis and Correction*) eram muito simples, mas já apresentavam ganhos materiais e humanos no processo de produção (Pietrosanto e Betta, 2000). Na indústria, as primeiras aparições, com essa denominação, datam de 1970 (Korbicz e Witczak, 2005). Com os avanços tecnológicos, os processos de produção se tornam cada vez mais complexos e, conseqüentemente, aumenta a demanda por sistemas de supervisão e proteção mais eficientes. Neste contexto, os modelos computacionais ditos “inteligentes” ganham destaques por apresentar “habilidades de aprendizado” e adequação às características intrínsecas do problema, principalmente, os modelos baseados em dados históricos da aplicação.

### 2.1.1 Supervisão e Proteção em Processos Industriais

Para Isermann e Balle (1997), o sucesso de um sistema de FDDC depende da qualidade da “supervisão” e “proteção” oferecidas pelo sistema como um todo. Na indústria, o objetivo principal da supervisão é evitar danos e acidentes. Para isso, é feito o monitoramento que determina de forma contínua e em tempo real o estado atual de funcionamento do sistema, incluindo, os estados não desejados e não permitidos. Os desvios do comportamento normal são distúrbios que podem resultar em eventos de falhas, ou de mau funcionamento do sistema ou processo. Também é função da supervisão a manutenção da operação mesmo na presença de falhas. Já a proteção representa a extinção de um comportamento indesejável e perigoso para a continuidade de funcionamento do “sistema físico” supervisionado. De modo ilustrativo, a Figura 2.1 (Adaptação de Andrade (2012)) mostra o esquema de um processo industrial supervisionado por um sistema de FDD acoplado, onde  $y$  é a variável de processo e  $\mu$  é o valor de referência.



**Figura 2.1:** Processo Industrial com FDD Acoplado.

Para uma melhor compreensão dos processos de supervisão e proteção nos sistemas de FDDC, algumas classificações e definições importantes, tais como: “estados”, “funções”, “modelos” e “propriedades”, postas por D’Angelo *et al.*, (2011), Isermann e Ballé, (1997), Sartori *et al.*, (2011) e Silva, (2008), são apresentadas a seguir.

Os “estados” são sinais captados e apresentados por instrumentos de medições, dentre os quais se destacam:

- **Faltas** - interrupção permanente de uma habilidade do sistema de executar uma função requerida sob condição de operação especificada (D’Angelo *et al.*, 2011);
- **Falhas** – desvio inesperado de uma propriedade característica ou de um parâmetro da sua

condição usual ou aceitável que tende a degradar o desempenho geral do sistema Isermann e Ballé (1997). Para Sartori *et al.* (2011), uma falha é qualquer desvio de uma característica (de um sistema/processo) em relação aos seus requisitos, podendo, ou não, afetar a capacidade de desempenhar uma função requerida (ocasionar um evento de falha ou um evento de mau funcionamento);

- Mau funcionamento - apresentação de irregularidades na funcionalidade do sistema;
- Erro - desvio entre um valor medido ou calculado (de uma variável de saída) e o valor verdadeiro/especificado/teoricamente correto;
- Distúrbio - entradas desconhecidas e não controladas para o sistema;
- Perturbação - modificação temporária do estado atual provocada por entradas que têm poder de ação no sistema;
- Resíduo - desvio entre o valor de entrada e a saída do modelo matemático utilizado, comprova a existência ou não da falha;
- Sintoma - mudança continuada e observável do comportamento normal.

As “funções” são atividades desenvolvidas pelos sistemas para a detecção e o diagnóstico dos sinais (D’Angelo *et al.*, 2011):

- Detecção de falha - determinação da presença de falhas em um sistema num dado instante de tempo;
- Isolamento de falha - determinação do tipo, da localização e do tempo da ocorrência de uma falha. Feita após a detecção de falha;
- Identificação da falha - determinação do tamanho e do comportamento, no tempo, de uma falha. Feita após a isolação (ou isolamento) de falha;
- Diagnóstico de falha - determinação do tipo, do tamanho, da localização e do tempo de uma falha. Feita após a detecção de falha. Inclui a isolação e a identificação de falha;
- Monitoramento - verificação contínua e em tempo real das variáveis mensuráveis do sistema, incluindo os estados não desejados e não permitidos e gerando alarmes quando os limites são ultrapassados;
- Supervisão - monitoramento e tomada de decisões apropriadas no intuito de manter a funcionalidade do sistema, mesmo na presença de falhas. A supervisão compreende as etapas de geração de atributos, detecção, avaliação, diagnóstico e classificação de falhas e tomada de decisões para a manutenção ou parada do sistema;
- Proteção - extinção de um comportamento perigoso de um sistema físico. A proteção automatizada inicia uma ação de segurança que mantém a funcionalidade do sistema ou o desligamento se houver risco eminente ao processo ou ambiente.

Os “modelos” são abordagens diferenciadas dos sistemas e podem ser classificados como:

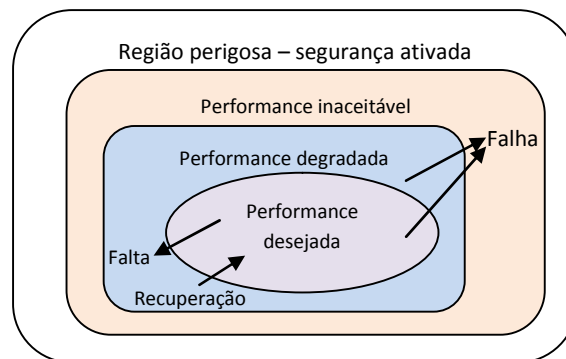
- Modelo Quantitativo - uso de relações estáticas e dinâmicas entre variáveis e parâmetros do sistema a fim de descrever o comportamento dos sistemas em termos matemáticos;
- Modelo Qualitativo - uso de relações estáticas e dinâmicas entre variáveis e parâmetros do sistema a fim de descrever o comportamento dos sistemas em termos qualitativos, tais como relações causas/ efeitos através de regras se/ então, lógica *fuzzy*, dentre outras;

- Modelo de Diagnóstico - um conjunto de relações estáticas ou dinâmicas que ligam variáveis específicas da entrada (sintomas) às variáveis específicas de saída (falhas).

As “propriedades do sistema” são habilidades apresentadas, dentre as quais se destacam:

- Confiabilidade - habilidade de um sistema de executar uma função requerida em condições indicadas por um período de tempo considerado;
- Disponibilidade - condição satisfatória de um equipamento ou sistema funcionar efetivamente em qualquer período de tempo;
- Segurança - habilidade de um sistema de não causar perigo para pessoas, equipamentos ou ambiente.

A Figura 2.2 mostra relações existentes entre alguns desses conceitos com riscos eminentes para um sistema ou processo de produção.

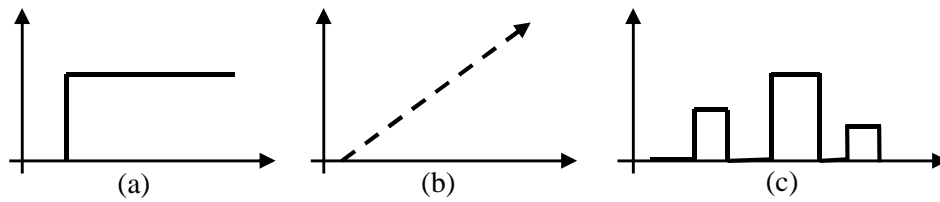


**Figura 2.2:** Conceitos Relacionados à Faltas e Falhas [adaptado de Silva (2008)].

Para Isermann (2006) a “falta” refere-se a um estado não desejado e com um leve desvio da condição normal do processo, embora a funcionalidade do sistema seja mantida. Já o termo “falha” corresponde a um estado não permitido e, conseqüentemente, se faz necessária a interrupção da funcionalidade do sistema. Independentemente de manutenção ou interrupção do processo, faltas e falhas são estados indesejados e ocorrem de forma inesperada. Por isso, os sistemas de FDDC devem apresentar habilidades de tolerância, de modo a evitar que faltas sejam transformadas em falhas. Isermann e Ballé (1997) definem uma falha como um desvio inesperado de um parâmetro ou propriedade característica do comportamento normal ou aceitável, mas que tende a degradar o desempenho geral do sistema. Para Sartori *et al.* (2011), a falha é o comprometimento parcial ou total dos requisitos de uma característica qualquer do sistema ou processo com a ocorrência do evento de mau funcionamento ou evento de falha, com a possibilidade de afetar o desempenho de uma função requerida.

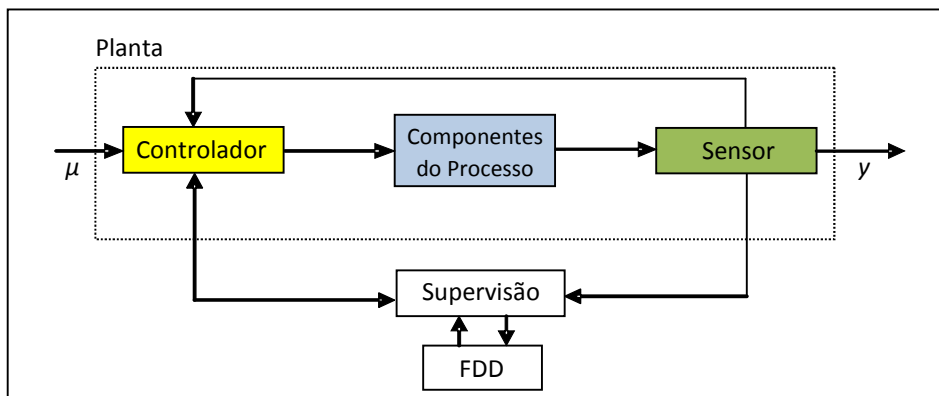
Na supervisão de processos industriais, as falhas são avaliadas e classificadas em diversos aspectos, dentre os quais se destacam o “tempo” e a “localização”. Na questão “temporal”, as falhas podem ser classificadas como abruptas, incipientes ou intermitentes, conforme descrição abaixo e na Figura 2.3:

- Falhas abruptas - resultam em grandes desvios nas condições normais de operação ou variáveis do processo e acontecem em um curto espaço de tempo;
- Falhas incipientes - afetam gradualmente o funcionamento normal do processo e, geralmente, levam um tempo maior para serem detectadas;
- Falhas intermitentes - alternam com o passar do tempo e são causadas por perturbações periódicas ou cíclicas, caracterizando um cenário de falhas.



**Figura 2.3:** Evolução Temporal das Falhas – (a) Falhas Abruptas; (b) Falhas Incipientes e (c) Falhas Intermitentes [Adaptado de Andrade (2012)].

Quanto ao “local” de ocorrência, as falhas podem acontecer nos atuadores, componentes do sistema ou nos sensores (Chiang *et al.*, 2001; Frank, 1990; Venkatasubramanian *et al.*, 2003). Essas falhas são mostradas na Figura 2.4 e descritas no que se segue:



**Figura 2.4:** Classificação de Falhas quanto à Localização [Adaptado de Andrade (2012)].

- Falhas em atuadores ou sensores - ocorrem por mau funcionamento dos atuadores ou sensores. Essas falhas configuram erros provocados por falhas físicas ou de equipamentos, desvios contínuos ou leituras fora dos padrões estabelecidos;
- Falhas no sistema dinâmico ou componente(s) do processo - são mudanças inaceitáveis nos parâmetros do processo, provocado por perturbações externas que alteram a relação original de entrada e saída do sistema. Como exemplo, alterações no coeficiente de transferência térmica de um trocador de calor devido a uma incrustação.

De acordo com as definições acima, Isermann (2006) define um sistema FDDC como um conjunto de técnicas utilizadas na supervisão de um processo para monitorar seu comportamento, prover informação acerca do mau funcionamento de seus componentes e permitir ações apropriadas para a manutenção da operação de modo a evitar perdas e acidentes. De forma similar, Sartori (2010) define um sistema FDDC como um conjunto de técnicas com o objetivo de estudar e desenvolver formas de assegurar o sucesso das operações planejadas através do reconhecimento de anormalidades no comportamento do processo e de ações de correção. Com essas definições apresentadas por Isermann (2006) e Sartori (2010), o atendimento desses requisitos técnicos é contemplado nas seguintes etapas:

- a) Detecção - compreende a determinação da presença de falhas no sistema e do tempo de detecção;
- b) Isolamento (classificação) - vem após a detecção e compreende a determinação do tipo, localização e tempo de ocorrência da falha;
- c) Identificação (análise) - compreende a determinação do tamanho, da causa e do comportamento da falha com o tempo, avaliando a origem da falha e o seu impacto no

sistema. As etapas de isolamento e identificação são englobadas pelo termo diagnóstico na sigla FDDC;

- d) Correção (ou recuperação) - refere-se à tomada automática, ou não, de ações apropriadas para restabelecer o funcionamento esperado do sistema e/ou evitar possíveis perdas humanas, ambientais ou econômicas.

As etapas acima realizam a supervisão e monitoramento de um sistema de FDDC, provendo informações das condições de funcionamento dos componentes e ações apropriadas de manutenção operacional do sistema. Porém, as ações em nível de correção do sistema podem estar além dos interesses e condições do pesquisador. Além disso, para Isermann e Ballé (1997), num sistema de FDD a “detecção” consiste na determinação da presença de um comportamento indesejável ou inaceitável do sistema. O “diagnóstico” compreende no isolamento (classificação) e identificação (análise) das falhas detectadas. Nessa subdivisão, o isolamento da falha se volta para avaliação das características mais objetivas, tais como: tipo, localização e tempo de ocorrência das falhas. Enquanto na identificação são analisadas as características mais complexas, tais como: tamanho, causa e comportamento temporal, origem e impactos no sistema. Como os sistemas de FDD propostos neste trabalho realizam apenas a detecção e a classificação dos estados de operação considerados nas aplicações, eles podem ser definidos como sistemas de detecção e isolamento de falhas (FDI - *Fault Detection and Isolation*).

## 2.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA), também conhecidas como sistemas conexionistas, computação neural, são atualmente uma área com fronteiras que vão das engenharias às ciências cognitivas. Inspiradas, inicialmente, na funcionalidade dos neurônios biológicos, a ideia básica das redes neurais é que unidades de processamento simples, dispostas em uma ou mais camadas, interagindo umas com as outras e com o ambiente, podem ter elevado poder computacional. As unidades de processamento recebem o nome de “neurônios”, também conhecidos como nodos ou neurônios artificiais, e a estrutura formada pelas unidades e interconexões entre elas levam o nome de “rede neural” ou “rede neural artificial”. Por meio de um processo de “treinamento”, as redes neurais passam a ser capazes de realizar atividades ditas “inteligentes”, mesmo que os dados utilizados nesse treinamento sejam não lineares, incompletos ou, até mesmo, contraditórios. A “habilidade” em manipular esses dados imprecisos faz com que as redes neurais sejam extremamente eficazes em tarefas onde os especialistas não estão à disposição, ou quando um conjunto de regras não pode ser facilmente formulado.

Formalmente, Hecht-Nielsen (1990) define uma RNA como uma estrutura com processamento de informação distribuída e paralela na forma de um grafo direcionado, com as seguintes subdefinições e restrições: (a) os nodos no grafo são unidades de processamento denominadas de neurônios; (b) as arestas são denominadas de “conexões/sinapses”, com sinais numa única direção; (c) cada unidade de processamento pode receber qualquer número de conexões de entrada, e também possuir qualquer número de conexões de saída, desde que os sinais nas conexões de saída sejam os mesmos; (d) as unidades de processamento podem ter memória local; (e) cada unidade de processamento possui uma função de transferência, a qual pode usar os valores armazenados na memória local e os sinais de entrada para produzir o sinal das saídas da unidade de processamento; e (f) os sinais de entrada para uma rede neural são conexões do ambiente externo ou de realimentação. As saídas da rede para o mundo externo são conexões que deixam a rede.

A capacidade de extrair conhecimentos, mediante a apresentação de exemplos representativos do problema, torna as redes neurais adequadas para a resolução de problemas do mundo real.

Esses exemplos apresentados à rede recebem o nome de padrões. O conjunto de pontos de uma imagem pode ser um padrão visual, do mesmo modo que uma sequência de fonemas pode ser um padrão auditivo. O primeiro exemplo é um padrão estático, para o qual não há qualquer informação temporal a ser considerada. O segundo é um padrão temporal, para o qual a informação temporal é considerada pela sequência dos seus componentes ou por sua velocidade de apresentação, embora cada “instância” de um padrão temporal possa ser considerada como um padrão estático. Além disso, uma parte de um padrão, visualizada por uma unidade de processamento, continua sendo um padrão.

As RNA, usualmente, operam em duas fases: a fase de “aprendizagem”; e a fase de uso. Na fase de aprendizagem, a rede é ajustada para produzir uma saída desejada em resposta ao meio ambiente, o que demonstra a “habilidade de aprender por experiência”. Na fase de uso, a resposta da rede pode ser, até certo ponto, insensível a pequenas variações que ocorram nas suas entradas. Esta característica é essencial no reconhecimento de padrões em ambientes do mundo real, nos quais a presença de distorções e ruídos é inevitável. Assim, quando um padrão de entrada, não visto durante o treinamento, é apresentado à rede, esta procura por uma entrada similar, aprendida na fase de treinamento, e produz uma saída semelhante à produzida por essa entrada já aprendida. Esta é a capacidade de generalização das redes neurais. O critério de similaridade, que define a semelhança de um exemplo aprendido e uma entrada vista pela primeira vez, deverá fazer parte do processo de aprendizagem. O conjunto de padrões para o treinamento deve ser então cuidadosamente escolhido, a fim de refletir o comportamento correto da rede durante a fase de uso. A capacidade de aprender por meios de exemplos e de generalizar a informação aprendida é, sem dúvida, a característica principal das redes neurais. E, para isso, as RNA são treinadas com algoritmos de aprendizado.

Denomina-se algoritmo de aprendizado um conjunto de regras bem definidas para a solução de um determinado problema. Os algoritmos de aprendizagem podem variar de acordo com os diferentes modelos de redes neurais, principalmente pelo modo como os pesos são modificados (Braga *et al.*, 2007). Nesse contexto, os paradigmas de aprendizado mais usados são do tipo supervisionado ou não supervisionado. No aprendizado supervisionado, existe a presença de um vetor alvo representando uma saída desejada. Durante a fase de treinamento, a rede é ajustada pelo algoritmo de aprendizagem para melhorar sua saída. O paradigma de aprendizado supervisionado inclui também um caso especial de aprendizado por reforço. Neste paradigma, não há a informação da saída desejada, e o desempenho é baseado na afirmação da saída estar correta ou incorreta. No paradigma não supervisionado, pelo fato de não haver o vetor de saída desejada, o algoritmo terá que buscar, nos padrões das entradas da rede, características comuns para que ela se auto-organize em classes semelhantes e, à medida que padrões diferentes são apresentados à rede, estes serão organizados nas diferentes classes existentes.

Em geral, novos algoritmos de aprendizagem surgem associados a novas arquiteturas ou a novos modelos de neurônios e *vice-versa*. Atualmente, existe uma variedade significativa de neurônios (Aleksander, 1967; Kan e Aleksander, 1987; Rumelhart e McClelland, 1986), mas essencialmente derivados de dois neurônios básicos: o modelo de “*McCulloch-Pitts*” (MCP) (McCulloch e Pitts, 1943), o qual simula de modo simplificado o neurônio natural e conhecido como neurônio “com pesos” em função da ponderação que faz sobre as suas entradas; e o modelo “digital”, sem nenhuma relação com o neurônio natural é também referenciado como neurônio/nodo “sem pesos”. Esses modelos neurais com e sem pesos são descritos com mais detalhes nas seções 2.2.1 e 2.2.2, respectivamente.



## 2.2.1 Redes Neurais Artificiais com Pesos

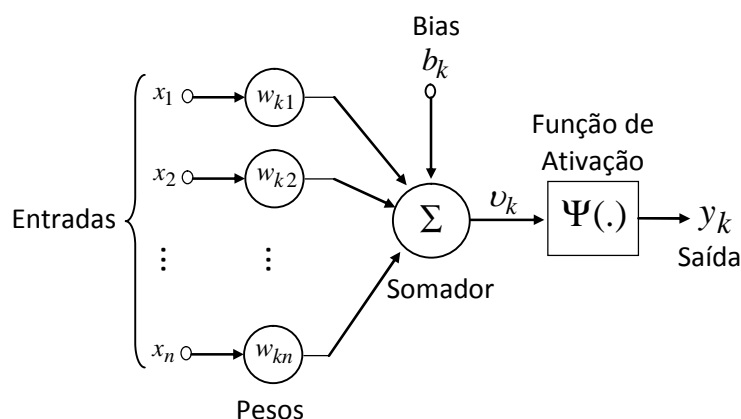
McCulloch e Pitts (1943), no trabalho intitulado “*A logical calculus of the ideas immanent in nervous activity*”, propuseram um modelo matemático, simplificado, do comportamento do neurônio biológico. O modelo foi chamado de neurônio “*McCulloch-Pitts*” (MCP) e tem como princípio o fato de, em um dado instante de tempo, o neurônio disparar ou permanecer inativo, o que pode ser entendido como verdadeiro ou falso. Assim, o neurônio artificial foi definido como um dispositivo binário, no qual o processamento das entradas ativa uma saída booleana 0 ou 1. Numa analogia com o neurônio biológico, as conexões e os pesos associados emulam os dendritos e sinapses, o mapeamento das entradas pelos pesos sinápticos emula o corpo celular e a função de saída emula o axônio.

O neurônio *McCulloch-Pitts* é usado na maior parte dos trabalhos com RNA e pode ser também referenciado de “neurônio com pesos”, em função da ponderação que ele faz sobre as suas entradas. A Figura 2.5 corresponde a um neurônio com pesos típico, descrito pelas Equações 2.1 e 2.2, as quais representam o produto escalar entre o vetor de pesos e o vetor de entradas e uma função de ativação não linear, respectivamente.

$$v_k = \sum_{i=1}^n w_{ki}x_i + b_k \quad (2.1)$$

$$y_k = \frac{1}{1 + \exp(-\alpha v)} \quad (2.2)$$

onde  $x_i$  são as entradas,  $w_{ki}$  são os pesos sinápticos associados às conexões em um neurônio  $k$ ,  $v_k$  é a ativação do  $k$ -ésimo neurônio,  $\alpha$  controla a forma da função sigmoide, e  $y_k$  é a saída para o  $k$ -ésimo neurônio. Os pesos  $w_{ki}$  e o bias  $b_k$  são os parâmetros que são ajustados pelos algoritmos de treinamento. A função de ativação, usada na Figura 2.5, pode ser substituída por outras funções de ativação (p.e., pela função *threshold*, pela tangente hiperbólica, entre outras). A preferência por funções de ativação sigmoides se dá pelo fato dessas funções serem contínuas, não lineares e diferenciáveis. Essas características permitem a utilização de métodos baseados em gradiente descendente, a exemplo do algoritmo *backpropagation* (“propagação para trás”) (Rumelhart *et al.*, 1986b), nos algoritmos de aprendizagem. Os modelos neurais compostos por nodos com pesos são chamados de “redes neurais com pesos”.



**Figura 2.5:** Um Neurônio com Pesos Típico (Adaptado de Oliveira, 2002).

Uma classe de redes neurais com pesos muito usada é constituída pelas redes *perceptrons* multicamadas, ou “*MultiLayer Perceptron*” (MLP). Informalmente, uma *multilayer perceptron* é uma rede neural artificial com pesos multicamada *feedforward* (Rumelhart *et al.*, 1986a). Uma

MLP consiste basicamente de uma camada de entrada, cujos neurônios possuem função de ativação linear, uma ou mais camadas escondidas (intermediárias) de neurônios com funções de ativação não lineares e uma camada de saída, com funções de ativação lineares ou não. Redes MLP “aprendem” relações lineares e não lineares existentes entre os valores de entrada e saída. Se a camada de saída for composta por neurônios lineares, a rede permite saídas com valores fora do intervalo  $[0, 1]$ . Porém, se os neurônios usarem funções de ativação não lineares do tipo sigmóides, a saída da rede ficará restrita aos valores contidos no intervalo  $[0, 1]$ .

As redes MLP são computacionalmente mais poderosas do que as redes sem camadas intermediárias. A solução de problemas não linearmente separáveis só foi possível por redes com uma ou mais camadas intermediárias (Braga *et al.*, 2007; Cybenko, 1988). Redes MLP com, no mínimo, uma camada intermediária e com neurônios lineares na saída, são teoricamente capazes de realizar a aproximação de qualquer função matemática, sendo ainda bastante utilizadas na resolução de problemas de associação e classificação de padrões (Cybenko, 1989). Existem diferentes modelos de algoritmos de aprendizagem que podem ser usados para o treinamento das redes MLP (Battiti, 1992; Hagan e Menhaj, 1994; Rumelhart e McClelland, 1986; Pearlmutter, 1992; Riedmiller, 1994). Esses algoritmos são geralmente do tipo supervisionado. O algoritmo de aprendizado mais conhecido para treinamento das redes MLP é o *backpropagation* (Rumelhart e McClelland, 1996). A formalização do algoritmo *backpropagation* contribuiu significativamente para o ressurgimento do interesse e expansão de trabalhos envolvendo redes neurais artificiais.

### 2.2.2 Redes Neurais Artificiais sem Pesos

Aleksander (1967), um dos pioneiros na área de RNA, propôs um modelo de neurônio/nodo totalmente digital, baseado em dispositivos de memória de acesso randômico ou aleatório, RAM (*Random Access Memories*). O modelo digital, inicialmente chamado de SLAM (*Stored Logic Adaptive Microcircuit*, microcircuito adaptativo de lógica armazenada) (Bledsoe e Browning, 1959) e posteriormente conhecido como RAM (Aleksander, 1966), deu origem a uma classe de neurônios baseados na lógica booleana, referenciados como neurônios “sem pesos” (*weightless neurons*) ou neurônios baseados em RAM (*RAM-based neurons*). As redes neurais compostas por esses neurônios são denominadas de “Redes Neurais Sem Pesos” (RNSP). Um nodo RAM e uma RNSP são formalmente definidos nas seções 3.2, 4.2.1.1 e 5.2.1.

A principal diferença entre as RNA que seguem o modelo convencional (com pesos) e as RNSP é a forma de armazenar informações. No modelo com pesos, as informações são armazenadas nos pesos associados às conexões entre os neurônios, enquanto que nas RNSP as informações são armazenadas em “memórias” inseridas nos próprios neurônios, em forma de tabelas-verdade.

O aprendizado nas redes neurais sem pesos consiste em mudanças nos conteúdos armazenados nos endereços de memória das RAM. Geralmente, os modelos neurais com pesos executam as etapas de treinamento, validação e teste. Nos modelos sem pesos, somente as etapas de treinamento e teste são executadas. O teste é realizado assim que o treinamento é finalizado e em muitos trabalhos é referenciado como validação.

Um neurônio com pesos consegue generalizar e computar funções linearmente separáveis (Rosenblatt, 1958). Já um neurônio sem pesos computa todas as funções booleanas de suas entradas, mas não consegue generalizar. Isto acontece porque uma única RAM na rede aprende ou reconhece apenas parte dos padrões submetidos à rede para aprendizado ou teste, conforme será mais detalhado na seção 3.2. Os subpadrões colocados como entradas para as RAM correspondem aos endereços acessados e são definidos por algum critério de mapeamento do

padrão de entrada (atributo qualitativo ou quantitativo) binarizado para a rede, o que também será mais detalhado na seção 3.2. Por isso, a generalização nas RNSP só acontece em nível da rede (Ludermir, *et al.*, 1999).

Uma característica comum das redes sem pesos é a baixa conectividade de cada neurônio aos demais neurônios da rede ou ao vetor de entrada. Isto é devido ao crescimento exponencial do espaço de memória requerido na razão direta do número de terminais de entrada, por neurônios. Assim, pode-se dizer que as redes sem pesos são, de modo geral, parcialmente conectadas, em contraste com outros tipos de redes completamente conectadas (Ludermir, *et al.*, 1999).

A praticidade e flexibilidade das RNSP, associadas às habilidades de aprendizagem, velocidade no processamento, eficiência em recuperação de informações, imunidade a ruídos, citadas por Yáñez e Camacho (2005) e confirmadas pelos sistemas de FDD aqui propostos, comprovam o bom desempenho das RNSP no reconhecimento e classificação de padrões. Ainda, as RNSP apresentam outras características importantes, tais como: proximidade com os sistemas digitais convencionais; algoritmos de aprendizagem rápidos e flexíveis; precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento das redes; neurônios sem pesos com entradas contínuas têm a capacidade natural de generalização (Clarkson *et al.*, 1992); redes cujos nodos lidam com entradas contínuas e aprendizado por reforço apresentam habilidades de aproximadores universais de funções contínuas (Gorse *et al.*, 1997). Também, a capacidade em aprender e reaprender, uma vez que as informações podem ser escritas, apagadas e reescritas na memória do dispositivo, equipara as RNSP, em poder computacional, com as Máquinas de Turing (de Oliveira, 1992).

Nesta tese, os estudos se concentram nos modelos neurais sem pesos. O desenvolvimento do nodo digital, baseado em dispositivos de memória de acesso aleatório foi, sem dúvida, o passo inicial para o desenvolvimento de muitas outras pesquisas tanto no nível teórico como no nível prático para a computação neural sem pesos. O dispositivo WiSARD (*Wilkie, Stonham e Aleksander's Recognition Device*, dispositivo de reconhecimento de Wilkie, Stonham e Aleksander), constituído de neurônios RAM, foi certamente um dos mais importantes resultados dessas pesquisas. O WiSARD foi disponibilizado comercialmente no início da década de 1980 e continua em uso em muitos trabalhos relacionados a problemas de reconhecimento de padrões. Depois do neurônio RAM, outros modelos de nodos sem pesos foram propostos na literatura, assim como o *Probabilistic Logic Neuron* (PLN) (Kan e Aleksander, 1987), o *Multiple-valued Probabilistic Logic Neuron* (MPLN) (Myers e Aleksander, 1988; Myers e Aleksander, 1989), e o nodo *probabilistic RAM* (pRAM) (Gorse e Taylor, 1988; Gorse e Taylor, 1991). Esses neurônios são apresentados e definidos com mais detalhes nos capítulos 3, 4 e 5. Os sistemas de FDD apresentados nos capítulos 3, 4 e 5 têm como base o dispositivo WiSARD implementado com neurônios RAM, MPLN e pRAM, respectivamente.

## **2.3 Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos**

### **2.3.1 Reconhecimento de Padrões**

O reconhecimento de padrões compreende a associação de elementos de um dado conjunto, com outros conjuntos previamente definidos e rotulados, conferindo-lhes uma identidade como classe (Duda e Hart, 1973). O processo consiste de tentativas com acertos e erros em busca de uma classificação ou partição conveniente de uma amostra de dados em grupos de elementos com características comuns - clusterização. Para esses autores, o reconhecimento de padrões é uma atividade de interesse tecnológico com regularidades significativas em ambientes ruidosos

ou complexos. A área de reconhecimento de padrões é tida como uma ciência não exata, devido à falta de padronização ou inconsistência nos dados, o que dificulta a identificação e classificação/clusterização dos dados avaliados (Duda e Hart, 1973).

Neste contexto, a “detecção” de falhas pode ser entendida como um problema de classificação, no qual um padrão de entrada  $x = [a_1, a_2, \dots, a_n]$  é considerado como estado de operação “normal” ou de “falha”. O “diagnóstico”, por outro lado, consiste na partição do estado de falha em subclasses com rótulos definidos pelos diferentes tipos de falhas considerados. Para isso, os atributos  $a_1, a_2, \dots, a_n$  devem conter informações que possibilitem a classificação desejada (Soares e Galvão, 2010). Formalmente, dado um padrão de entrada  $x$  e uma variável de resposta  $y$ , busca-se uma aproximação para a relação  $y = f(x)$ . Nessa relação  $y$  é um rótulo de classe com valor no conjunto  $\{C_1, C_2, \dots, C_k\}$ , com  $k \in \mathbb{N}$  e a função  $f(x)$  define, de forma aproximada, os limites para as classes consideradas. Para uma melhor compreensão da importância da pesquisa realizada, a seção seguinte apresenta o estado da arte sobre detecção e diagnóstico de falhas com uso das redes neurais sem pesos.

### 2.3.2 Detecção e Diagnóstico de Falhas com Redes Neurais sem Pesos

Conforme já mencionado anteriormente, a aparição dos primeiros sistemas de detecção e diagnóstico de falhas na indústria data de 1970 (Korbicz e Witczak, 2005). Coincidentemente, os primeiros trabalhos com redes neurais sem pesos apareceram, basicamente, no mesmo período, mais precisamente no final da década de 1960 (Alexandre, 1966, 1967). Também, o reconhecimento e a classificação de padrões é o “cerne” dessas duas linhas de pesquisas.

Conforme também já mencionado anteriormente, para Isermann e Ballé (1997) num sistema de FDD a “detecção” consiste em determinar a presença de um comportamento indesejável ou inaceitável do sistema e o momento de sua percepção. Por outro lado, o “diagnóstico” compreende o isolamento (classificação) e a identificação (análise) das falhas detectadas. Neste contexto, os sistemas de FDD podem ir além do reconhecimento e classificação de padrões e serem considerados como sistemas complexos, principalmente por atuarem em ambientes dinâmicos. Já os sistemas neurais sem pesos eram voltados basicamente para o reconhecimento e classificação de imagens digitais, considerados, na grande maioria, como sistemas estáticos. Num processo de reconhecimento e classificação de padrões, um sistema é considerado dinâmico se os valores das saídas dependem de valores passados das entradas, caso contrário, se a saída é função apenas da entrada atual, o sistema é considerado estático (Birkhoff, 1927). No entanto, estudos mais recentes comprovam a aplicabilidade das RNSP em ambientes dinâmicos (Bandeira 2010; Cardoso *et al.*, 2011 e 2013; Cardoso, 2012; Oliveira *et al.*, 2017).

Com a revisão da literatura, foi possível notar um número significativo de estudos formais voltados para o problema de detecção e diagnóstico de falhas, principalmente em processos industriais. Esses estudos, na grande maioria, utilizam modelos computacionais ditos “inteligentes”, com destaque para as redes neurais artificiais com pesos, lógica *fuzzy*, algoritmos genéticos e redes bayesianas (Andrade, 2012; D’Angelo, *et al.*, 2011; Lemos *et al.*, 2010; Lemos, 2011; Linden, 2008; Medeiros, 2009; Maia *et al.*, 2009; Rivera, 2007; Sartori, 2012). Porém, com exceção do trabalho apresentado por Oliveira *et al.* (2017) e descrito no capítulo 3 desta tese, nenhum outro trabalho encontrado na literatura tratou o problema de FDD com uso das RNSP, principalmente com a terminologia “detecção e diagnóstico de falhas”. No entanto, muitos trabalhos encontrados utilizam as RNSP em diferentes aplicações, com predominância no reconhecimento e classificação de padrões, dentre as quais se destacam as descritas abaixo:

- A “navegação de robôs” (Bonato *et al.*, 2004; França *et al.*, 2010; McElroy e Howells, 2011; Nurmaini, *et al.*, 2009; Nurmaini e Tutuko, 2011; Yao *et al.*, 2003). O trabalho

apresentado por Yao *et al.* (2003) faz uso de uma rede neural baseada em memórias RAM para a navegação de um robô móvel, controlado por um sistema de microprocessadores simples para detectar e desviar de obstáculos em tempo real. Bonato *et al.* (2004) apresentaram um sistema de visão computacional incorporado a um robô móvel, com tecnologias reconfiguráveis. O referido sistema captura gestos por meio de uma câmera digital a cores e executa o reconhecimento em tempo real. Esse sistema serve de interface de comunicação humano-robô com aplicações diversas, a exemplo de pessoas com deficiência. Nurmaini *et al.* (2009) propuseram uma rede neural sem pesos para reconhecer e classificar ambientes visitados por um robô móvel. O sistema é composto por um conjunto de microprocessadores simples e demonstra eficiência no reconhecimento de obstáculos em tempo real pelo robô utilizado. França *et al.* (2010) propuseram um sistema de visão artificial baseado na rede neural booleana WiSARD e com objetivos de seguir a cadência de um navio em seis eixos de liberdade e em tempo real. McElroy e Howells (2011) utilizaram uma rede neural sem pesos e um algoritmo genético para manipular automaticamente entradas e saídas de dados. O sistema proposto permite que um robô possa guiar um veículo em situações reais e perigosas, onde a segurança é fundamental e a realização por seres humanos é desaconselhável. O algoritmo de reconhecimento de padrões apresentado por Nurmaini e Tutuko (2011) classifica ambiente e auxilia na tomada de decisões para a navegação de um robô móvel. A estratégia demonstrou bons resultados com um robô móvel alimentado por microcontroladores e sensores de baixo custo. Os resultados experimentais apresentaram taxas de reconhecimento dos ambientes de até 98,5%, além de evitar obstáculos em tempo real;

- O “reconhecimento de dígitos e impressões digitais” (Al-Alawi, 2007; Bandeira *et al.*, 2009; Cardoso, *et al.*, 2013; Conti, *et al.*, 2009; Grieco *et al.*, 2010). Al-Alawi (2007) apresenta um sistema de classificação padrão neuro-*fuzzy* baseado na arquitetura de rede neural sem pesos para o reconhecimento de dígitos manuscritos. O sistema de inferência *fuzzy* aprende regras *fuzzy* extraídas dos vetores de características gerados por uma rede neural sem pesos de uma única camada. Os resultados experimentais comprovaram um desempenho superior comparado com o desempenho de uma rede RAM tradicional. Bandeira *et al.* (2009) usaram uma rede neural sem pesos denominada de NC-WiSARD. A rede utiliza uma estrutura multicamadas e aprendizado não supervisionado, o autoWiSARD. O sistema é capaz de reconhecer as imagens treinadas, bem como versões modificadas delas. A rede NC-WiSARD consegue reconhecer e compor padrões mais complexos através de sua estrutura hierárquica. Conti *et al.* (2009) propuseram um sistema de identificação automática de impressões digitais baseado em redes neurais sem pesos. O trabalho realiza de forma rápida a classificação de impressões digitais com o uso de mapas direcionais para cinco classes distintas. Grieco *et al.* (2010) objetivam associar modelos de dígitos manuscritos com a formação de imagens mentais. Essa equivalência é obtida por meio de contagem de frequência e filtragem e ou por formulação de regras *fuzzy*, melhorando assim o processo de classificação da WiSARD. Cardoso *et al.* (2013) propuseram um sistema para posicionamento (pontos de acesso WiFi) de usuários em ambientes internos (museus, centros comerciais, fábricas, dentre outros) com o reconhecimento de impressões digitais por um o sistema neural sem pesos, baseado na rede WiSARD, com resultados considerados muito bons em termos de desempenho e de solução de posicionamento;
- O “reconhecimento de faces e impressões faciais” (Araújo, 2011; Sirlantzis *et al.*, 2009; Vidal, *et al.*, 2013). Sirlantzis (2009) propôs um sistema de reconhecimento biométrico facial com arquitetura em camadas modulares. Cada camada corresponde a uma rede

WiSARD, cujos discriminadores são considerados módulos/ classificadores. As RAM nos discriminadores armazenam probabilidades. A saída de um discriminador é a combinação das probabilidades de saídas das RAM com base na função soma (método de agregação mais usado em um sistema multiclassificador). O padrão de teste é reconhecido pelo discriminador com maior soma de probabilidade. Araújo (2011) desenvolveu um sistema denominado de RWiSARD com habilidades para o reconhecimento e classificação de imagens faciais em escala de cinza. O RWiSARD foi adaptado para lidar com valores escalares e com possibilidade de representação no campo dos reais. Vidal *et al.* (2013) aplicaram redes neurais sem pesos para o reconhecimento de emoções. O sistema proposto analisa imagens de expressões faciais com emoções do tipo: raiva; desprezo; nojo; medo; felicidade; neutro; tristeza; e surpresa. Os dados experimentais foram extraídos da base de dados TFEID (*Taiwanese Facial Expression Image Database*, banco de dados de imagem de expressão facial de Taiwan) e os resultados comprovam o bom desempenho da metodologia para a referida aplicação;

- O “agrupamento de fluxos de dados” (Cardoso *et al.*, 2011; Cardoso, 2012). Cardoso *et al.* (2011) propuseram um sistema neural sem pesos evolutivo para agrupamento de fluxos de dados em ambientes dinâmicos. A rede WiSARD usada realiza um aprendizado *one-shot* (rápido) e produz protótipos das categorias aprendidas. Cardoso (2012) propôs um sistema denominado de StreamWiSARD para o agrupamento de dados em fluxos contínuos. O sistema proposto cria dinamicamente a estrutura da rede com base no aprendizado não supervisionado e na rede autoWiSARD (Bandeira, 2010) e evidencia questões como temporalidade e velocidade na apresentação dos fluxos de dados;
- A “previsão de séries temporais” (Mpofu, 2006). Mpofu (2006) propôs um sistema de previsão de preços das ações no mercado usando uma rede neural sem pesos. O sistema proposto é testado com dados de estoque obtidos a partir do Reino *Financial Holdings* do Zimbábue e utiliza a suavização exponencial simples para a previsão realizada e faz a comparação com duas ferramentas de previsão usando técnicas quantificáveis.

A “ausência” de sistemas de FDD baseados nas RNSP e o “potencial” dessas redes para o reconhecimento e a classificação de padrões foram os principais fatores que motivaram o desenvolvimento dos sistemas de FDD propostos neste trabalho. Os sistemas desenvolvidos foram validados por três estudos de caso, apresentados na seção 2.5 e descritos com mais detalhes nos capítulos 3, 4 e 5.

## 2.4 Materiais e Métodos

O desenvolvimento deste trabalho tomou como pressuposto o caráter desordenado e não previsível do processo de construção de conhecimentos, na busca por uma produção ordenada, lógica e de contribuição acadêmica e científica. Neste contexto, as características qualitativas foram extraídas de medidas quantitativas obtidas pelos instrumentos de medições em processos reais ou simulados. De forma a contemplar toda a pesquisa, os materiais e métodos usados se complementam e se subdividem nas seguintes etapas:

- a) Revisão dos pressupostos teóricos sobre detecção, diagnóstico e correção de falhas, com ênfase na supervisão dos processos industriais;
- b) Revisão dos modelos computacionais, com ênfase nas redes neurais sem pesos;
- c) Compreensão da composição e funcionalidade das plantas industriais usadas para a

validação dos sistemas FDD propostos;

- d) Composição das bases de dados usadas para o treinamento e validação dos sistemas propostos;
- e) Proposição dos sistemas de FDD com base nas redes neurais sem pesos, na seleção de atributos, nos algoritmos de granulação dos atributos selecionados e filtros de agrupamentos para diagnósticos por grupos de falhas. Para cada sistema e aplicação é proposta uma configuração final como a de melhor desempenho para a detecção e o diagnóstico realizados;
- f) Desenvolvimento dos códigos computacionais relacionados a cada sistema de FDD proposto;
- g) Elaboração dos demais capítulos componentes desta tese, que finaliza o trabalho de doutoramento, fundamentados nos estudos teóricos e práticos desenvolvidos.

Para uma compreensão inicial dos trabalhos desenvolvidos nesta tese, as seções 2.5 e 2.6 seguintes apresentam, sucintamente, os três estudos de casos e os três sistemas de FDD propostos neste trabalho. Esses sistemas são detalhados nas seções 3.3, 4.3 e 5.3, respectivamente.

## **2.5 Estudos de Casos**

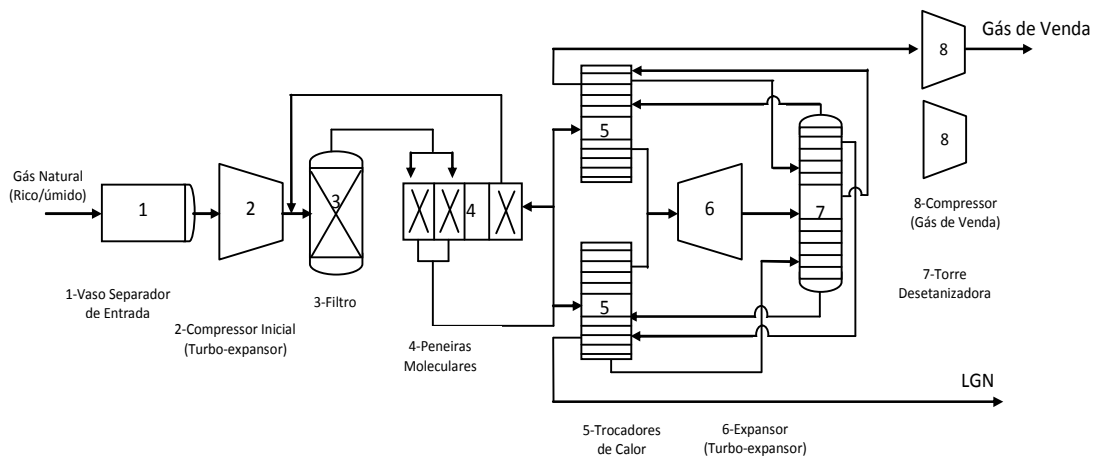
Conforme já mencionado anteriormente, os sistemas propostos neste trabalho foram validados por três estudos de caso. Um deles compreende em um problema real e univariável e os outros dois simulam o funcionamento de uma planta industrial da indústria química e de um reator tanque agitado continuamente, respectivamente. Esses estudos de casos são apresentados resumidamente nas três seções seguintes e com detalhes nas seções 3.4, 4.4 e 5.4.

### **2.5.1 Unidade de Processamento de Gás Natural (UPGN)**

Em sua cadeia produtiva, o gás natural pode ser associado ou não ao petróleo e varia de campo para campo de acordo com o tipo de matéria orgânica originária. O seu processamento é realizado por uma “Unidade de Processamento de Gás Natural” (UPGN) com função principal de separar os hidrocarbonetos leves, a exemplo do metano e etano, daqueles mais pesados, com destaque para butano, pentano, hexano, heptano e octano (Andrade, 2012). Os produtos finais com essa separação são o gás natural e o gás úmido na forma líquida, denominado de “Líquido de Gás Natural” (LGN). Essa separação pode ser realizada por métodos como refrigeração simples, absorção refrigerada, turbo-expansão e expansão Joule-Thompson, nos quais a condensação dos hidrocarbonetos mais pesados acontece com a diminuição da temperatura do gás processado (Andrade, 2012). Na UPGN em estudo, o método utilizado para o resfriamento do gás processado é a turbo-expansão, por apresentar as melhores taxas de separação das frações mais pesadas do gás industrial. Neste método a condensação da fração mais pesada acontece quando o gás é passado por um turbo-expansor com a troca de calor a baixas temperaturas (Vaz *et al.*, 2008).

O gás industrial processado é também chamado de gás combustível, gás seco, gás pobre, gás residual, gás especificado ou gás para a venda (Vaz *et al.*, 2008). Usado na indústria como fonte de calor, geração de energia e matéria-prima, com destaque para os setores químicos, petroquímicos e de fertilizantes. Já o LGN é também chamado de gás rico, do qual deriva, dentre outros, a gasolina natural e o butano, utilizado na produção do “Gás Liquefeito do Petróleo” (GLP) (ANP, 2014).

Como ilustrado na Figura 2.6, o processamento do gás natural na UPGN em estudo compreende as etapas de separação, desidratação, resfriamento e fracionamento (Sartori, 2012). Inicialmente, o processo de separação consiste na retirada da água e dos hidrocarbonetos líquidos do gás processado. O processo continua com a desidratação do gás que compreende a retirada da parte restante da água encontrada na forma de vapor. Com isso, o gás é secado e toda a parte líquida é associada aos hidrocarbonetos mais pesados. O resfriamento do gás natural acontece no turbo-expansor que chega a uma coluna de desetanização para o fracionamento. As alternativas de fracionamento dependem da composição do gás processado e do destino final dos produtos obtidos. Se o gás processado for do tipo associado, o gás industrial e o LGN são armazenados e, oportunamente, vendidos e comercializados para as Companhias Distribuidoras Estaduais de gás e para as refinarias, respectivamente (Vaz *et al.*, 2008).



**Figura 2.6:** Esquema do Processamento do Gás Natural da UPGN em Estudo [Adaptado de Sartori, 2012)].

Depois de processado o gás natural é uma mistura de hidrocarbonetos leves que, à temperatura ambiente e pressão atmosférica, permanece no estado gasoso. Trata-se de um gás inodoro e incolor, não tóxico e mais leve do que o ar. O gás natural é uma fonte de energia mais limpa que pode ser usada nas indústrias, substituindo outros combustíveis mais poluentes como óleos combustíveis, lenha e carvão. Na forma líquida deve ser mantido à temperatura de 160 °C e pressão próxima da atmosférica. Exige um complexo sistema de armazenamento e transporte específico. Na matriz energética apresenta vantagens ambientais se comparada com outras fontes fósseis (carvão mineral e derivados de petróleo) de energia, tais como (Hayashi, 2013): baixa presença de contaminantes; combustão mais limpa; redução do desmatamento; menor emissão de CO<sub>2</sub> por unidade de energia gerada; menor esforço no tratamento dos gases poluentes; facilidade de transporte e manuseio em relação a outros tipos de combustíveis; dentre outras.

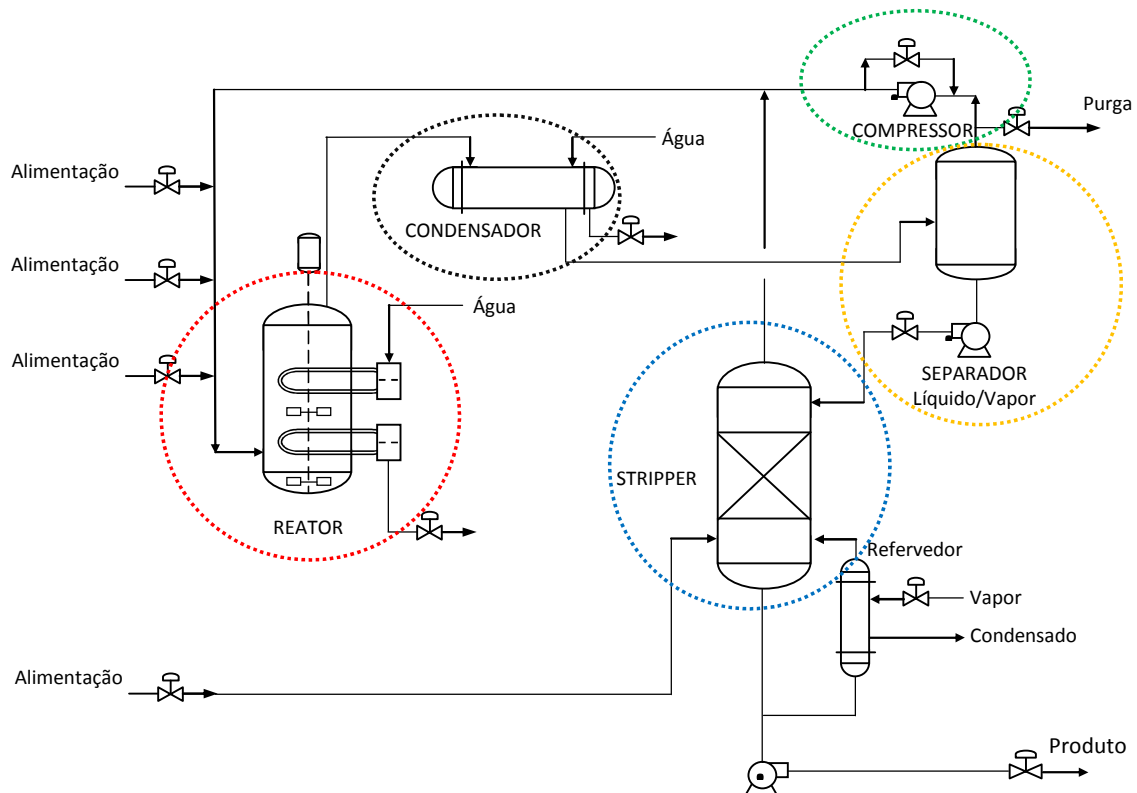
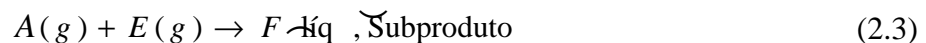
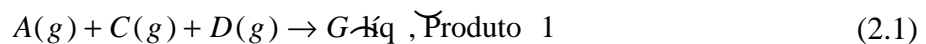
O Estado da Bahia possui três unidades de processamento de gás natural, uma no município de Candeias e duas em Pojuca. A unidade em estudo, Pojuca/ Bahia 3, possui uma capacidade nominal de 2,5 milhões m<sup>3</sup>/dia de gás industrial. Na estrutura funcional da unidade faz-se necessária a utilização de um potente compressor alternativo, chamado de compressor de gás para a venda (SGC, *Sale Gas Compressor*) operando com carga mínima de 50%. O monitoramento da temperatura do SGC por meio dos sensores acoplados a ele corresponde ao objeto de pesquisa para este estudo de caso. Conforme apontado por Sartori (2012), o sistema de compressão de gás para a venda contribui significativamente para as falhas ocorridas na planta da unidade, com destaque para as falhas por vibração do compressor. De acordo com Sartori (2012), a vibração do compressor de gás para a venda provoca o mau funcionamento



dos sensores ou até mesmo o rompimento de suas hastes de transmissão. O comprometimento de um sensor é suficiente para o acionamento indevido do sistema de intertravamento da planta, o que justifica o desenvolvimento de um sistema de detecção e diagnóstico de falhas que amenize o problema apontado. Este estudo de caso foi usado com o sistema FDD-WiSARD proposto no capítulo 3. Os detalhes sobre o SGC e a formação das bases de dados utilizadas para o treinamento e validação do referido sistema estão apresentados na seção 3.4.1.

### 2.5.2 Tennessee Eastman Process (TEP)

O *Tennessee Eastman Process* (TEP) simula um processo químico tido como referência (*benchmark*) em estudos sobre controle de processos e detecção e diagnóstico de falhas, por se caracterizar como um processo realista e com marcante não linearidade em seu comportamento. O modelo TEP foi proposto originalmente por Downs e Vogel (1993) como uma planta industrial em malha aberta, centrado em cinco unidades principais de operação, denominadas de “reator”, “condensador”, “separador líquido/vapor”, “compressor”, e “coluna de separação (*stripper*)”, em destaques na Figura 2.7 (adaptada de Soares e Galvão 2010). O TEP utiliza 12 variáveis manipuláveis e 41 variáveis medidas e objetiva a produção dos produtos *G* e *H* a partir dos reagentes *A*, *C*, *D* e *E*, gerando um subproduto *F*, com base nas Equações 2.1 a 2.4. Além dos reagentes e produtos é usado também um inerte gasoso *B*, introduzido nas correntes de alimentação dos reagentes.

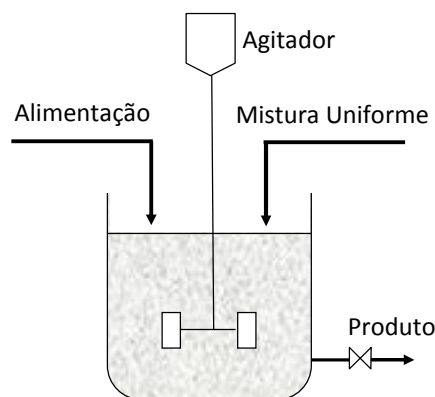


**Figura 2.7:** Diagrama Esquemático do *Tennessee Eastman Process* (TEP).

O modelo TEP original apresentado por Downs e Vogel (1993) possui 12 variáveis manipuláveis e 41 variáveis medidas. Com essas variáveis, os autores consideram seis modos de operação e 21 condições de falhas simulados em malha aberta. Mais detalhes do modelo TEP são apresentados nas seções 3.4.1, 4.4.1 e 5.4.1 dos capítulos 3, 4 e 5, respectivamente.

### 2.5.3 Continuous Stirred Tank Reactor (CSTR)

O reator tanque agitado continuamente (CSTR, *Continuous Stirred Tank Reactor*) é um tanque agitado com escoamento contínuo e sem acúmulo de reagentes ou produtos. Em um reator CSTR um ou mais reagentes fluidos são introduzidos no reator, equipado com um agitador, enquanto o efluente do reator é removido. O agitador mistura os reagentes assegurando uma homogeneização do sistema reativo. O reator aqui estudado (Singhal e Seborg, 2002) possui as seguintes características: não isotérmico e com resfriamento na jaqueta; comportamento dinâmico; nível de líquido variável; composição de saída igual à do interior do reator; taxa da reação igual em todo o reator, inclusive na saída; uso preferencial em problemas com agitação intensa e reações em fase líquida. O CSTR atende os princípios de mistura perfeita e de parâmetros físicos constantes e tem como base a reação irreversível de primeira ordem clássica do tipo A → B. Um diagrama geral e estrutural do CSTR é apresentado na Figura 2.8 (Adaptado de Singhal e Seborg, (2002)).



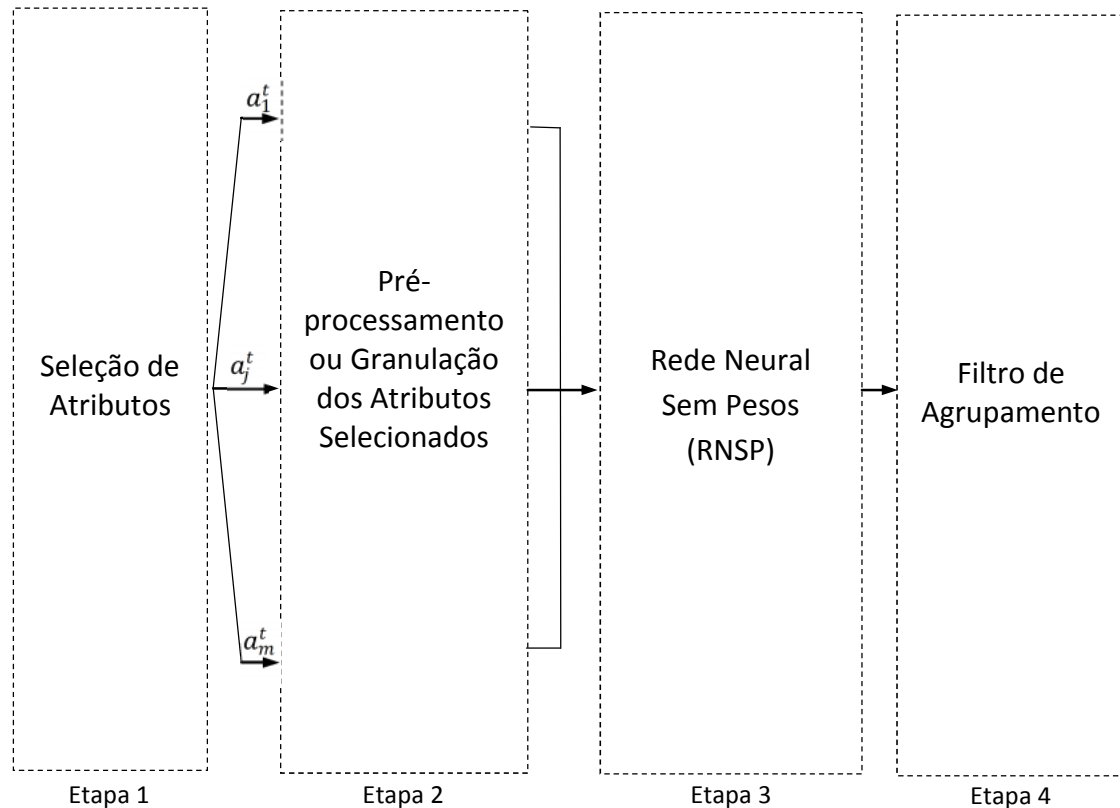
**Figura 2.8:** Diagrama Esquemático do Reator CSTR.

O simulador aqui utilizado tem como referência o trabalho apresentado por Singhal e Seborg (2002) e simula 18 dos 28 estados de operação considerados por esses autores. O referido simulador se encontra em processo de construção pelo grupo de pesquisa PROTEC da Universidade Federal da Bahia (UFBA), fato que justifica a consideração “parcial” dos estados de falhas considerados por Singhal e Seborg (2002).

No contexto dos estudos de detecção e diagnóstico de falhas há muitos trabalhos com uso dos modelos CSTR, a exemplo dos realizados por Vaidyanathan e Venkatasubramanian (1992) e Sorsa e Koivo (1993). O estudo realizado por Singhal e Seborg (2002) inclui o estado de operação normal e 27 condições de falhas, simulados por 14 variáveis do processo consideradas. Mais detalhes sobre a estrutura e funcionamento do reator considerado podem ser vistos nas seções 4.4.2 e 5.4.2. Ainda, um estudo mais aprofundado sobre as estruturas de controle e os parâmetros do controlador para o referido estudo de caso podem ser vistos em Singhal e Seborg (2002) e Johannesmeyer (1999).

## 2.6 Sistemas Propostos

A estrutura apresentada na Figura 2.9 contempla as estruturas e funcionalidades apresentadas pelos três sistemas FDD propostos. Esses sistemas são constituídos de quatro etapas distintas e complementares, denominadas de: “seleção dos atributos”; “pré-processamento dos atributos selecionados”; “rede neural sem pesos”; e “filtro de agrupamentos”. Como o estudo de caso UPGN é univariável, a etapa de seleção dos atributos não é executada.



**Figura 2.9:** Diagrama Estrutural para os Sistemas FDD Propostos.

Conforme mostrado na Figura 2.9, o sistema inicia com a seleção de atributos por classe, seguida do pré-processamento ou granulação dos atributos selecionados. As saídas do pré-processamento ou grânulos são padrões comportamentais que alimentam as redes neurais sem pesos, compostas por neurônios baseados em dispositivos de memórias RAM. A detecção e o diagnóstico de falhas por classe ou grupos de falhas são obtidos diretamente nas saídas das redes testadas ou por um filtro de agrupamento, como etapa final do sistema FDD. Esses três sistemas FDD são sucintamente descritos a seguir, e as suas estruturas e funcionalidades são apresentadas com detalhes nas seções 3.3, 4.3 e 5.3, respectivamente.

Sistema “FDD-WiSARD” (sistema 1/ capítulo 3). O sistema proposto executa, previamente, a seleção de atributos (variáveis) por classe e o mapeamento temporal dos dados e, na sua etapa intermediária a rede neural sem pesos WiSARD, com neurônios RAM padrão e contador, realiza a detecção e o diagnóstico por classe (Oliveira *et al.*, 2017). As saídas da rede são levadas a um filtro de agrupamento, na etapa final do sistema, se um diagnóstico por grupos de falhas for necessário. O sistema foi validado com os estudos de casos UPGN e TEP. Os resultados mostram o sucesso dos sistemas para a detecção e o diagnóstico de falhas com percentuais de acertos na classificação de 98,78% e 99,47% para as respectivas aplicações.

Sistema “FDD-RNSPG” (sistema 2/ capítulo 4). O sistema inicia com a seleção de atributos por classe, seguida da granulação dos atributos selecionados. A granulação dos dados é fundamentada na matemática intervalar e na lógica *fuzzy* (Leite *et al.*, 2009). As saídas dos grânulos criados alimentam redes neurais sem pesos, compostas por neurônios lógicos probabilísticos multivalorados (MPLN, *Multiple-valued Probabilistic Logic Neuron*) (Myers e Aleksander, 1988, 1989). A detecção e o diagnóstico de falhas por classe ou grupos de falhas são obtidos diretamente nas saídas das redes testadas ou por um filtro de agrupamento, como etapa final do sistema de FDD. O sistema foi validado com os estudos de casos TEP e CSTR. Os resultados mostram o sucesso do sistema para a detecção e o diagnóstico de falhas com percentuais de acertos na classificação de 99,85% (TEP) e 99,70% (CSTR) para as respectivas aplicações.

Sistema FDD-pRAM (sistema 3/ capítulo 5). O sistema inicia com a seleção das variáveis de maior relevância para cada classe considerada na aplicação. Os atributos selecionados são granulados por um sistema de inferência *fuzzy* recursivo baseado nos conceitos de *clouds* (“nuvens”) propostos em Angelov e Yager (2011) e no algoritmo usado por Rosa *et al.* (2013). Para cada atributo selecionado é gerado um conjunto de *clouds* e a cada padrão de entrada uma *cloud* é escolhida para cada atributo selecionado. As *clouds* selecionadas formam padrões comportamentais, os quais são apresentados à rede neural sem pesos com neurônios RAM probabilísticos (pRAM, *probabilistic RAM*) (Gorse e Taylor, 1988). A detecção e o diagnóstico de falhas por classe ou grupos de falhas são obtidos diretamente nas saídas das redes testadas ou por um filtro de agrupamento, como etapa final do sistema. O sistema FDD-pRAM foi validado com os estudos de casos TEP e CSTR. Os resultados mostram o sucesso do sistema para a detecção e o diagnóstico de falhas com percentuais de acertos na classificação de 99,89% e 99,88% para as respectivas aplicações.

## Abreviaturas

ANP:	Agência Nacional de Petróleo e Gás Natural.
autoWiSARD:	Unsupervised Modes for the WISARD.
CSTR:	Continuous Stirred Tank Reactor.
FDD:	<i>Fault Detection and Diagnosis.</i>
FDDC:	<i>Fault Detection, Diagnosis and Correction.</i>
FDD-pRAM:	Detecção e Diagnóstico de Falhas com RAM Probabilísticas.
FDD-RNSPG:	Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Granulares.
FDD-WiSARD:	Detecção e Diagnóstico de Falhas com Base no Dispositivo WiSARD.
FDI:	<i>Fault Detection and Isolation.</i>
GLP:	Gás Liquefeito do Petróleo.
LGN:	Líquido de Gás Natural.
MCP:	<i>McCulloch-Pitts.</i>
MLP:	<i>MultiLayer Perceptron.</i>
MPLN:	<i>Multiple-valued Probabilistic Logic Neuron.</i>
NC-WISARD:	Arquitetura multicamadas do Neocognitron com base na rede WiSARD.

Petrobras:	Petróleo Brasileiro S.A.
PLN:	<i>Probabilistic Logic Neuron.</i>
pRAM:	<i>probabilistic Random Access Memory.</i>
PROTEC:	Processos de Tecnologias.
RAM:	<i>Random Access Memory.</i>
RAM-based:	<i>RAM-based neurons.</i>
RNA:	Rede neural Artificial.
RNSP:	Rede Neural Sem Pesos.
RWiSARD:	Rede WiSARD com valores Reais.
SGC:	<i>Sale Gas Compressor.</i>
SLAM:	<i>Stored Logic Adaptative Microcircuit.</i>
StreamWiSARD:	Rede WiSARD auto organizável para agrupamento de fluxo de dados.
TEP:	<i>Tennessee Eastman Process.</i>
TFEID:	<i>Taiwanese Facial Expression Image Database.</i>
UFBA:	Universidade Federal da Bahia.
UPGN:	Unidade de Processamento de Gás Natural.
WiSARD:	<i>Wilkes, Stonham, Aleksander Recognition Device.</i>

## Referências

- Al-Alawi, R. (2007). Performance evaluation of fuzzy single layer weightless neural network. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15 (3), 381-393.
- Aleksander, I. (1966). Self-adaptive universal logic circuits. *Electronics Letters*, 2, 231.
- Aleksander, I. (1967). Adaptive systems of logic networks and binary memories. *In Proc. Spring Joint Computer Conference.*
- Andrade, V. E. (2012). Sistema de detecção e diagnóstico de falhas em sensores de um compressor de gás natural utilizando lógica fuzzy tipo-2. *Dissertação de Mestrado*, Programa de Engenharia Industrial. Universidade Federal da Bahia-UFBA, Salvador, Bahia, Brasil.
- ANP (2014). Boletim da produção de petróleo e gás natural – julho 2014. *Agência Nacional do Petróleo, Gás Natural e Biocombustíveis.*
- Araújo, L. A. (2011). *RWiSARD*: Um modelo de rede neural sem peso para reconhecimento e classificação de imagens em escala de cinza. *Dissertação de Mestrado*, Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.
- Bandeira, L. C., França, H. L., França, F. M. G. (2009). NC-WiSARD: Uma interpretação booleana da arquitetura Neocognitron. *In: Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional, CD-Rom*, Ouro Preto, MG, Brasil.

- Battiti, R. (1992). First and second-order methods for learning: between steepest descent and Newton's method. *Neural Computation*, 4(2), 141-166.
- Birkhoff, G. (1927) Dynamical Systems. Ed. AMS books.
- Bledsoe, W., Browning, I. (1959). Pattern recognition and reading by machine. *In Proc. Eastern Joint Computer Conference*, Boston, 225-232.
- Bonato, V., Sanches, A. K., Fernandes, M. M., Cardoso, J. M., do Valle Simões, E., Marques, E. (2004). A real time gesture recognition system for mobile robots. *International Conference on Informatics in Control, Automation and Robotics*, Portugal. 2, 207-214.
- Braga, A. P., Carvalho, A. P. L. F., Ludermir, T. B. (2007). Redes neurais artificiais: Teoria e aplicações. *Livros Técnicos e Científicos, Editora S.A, 2ª Edição*, Rio de Janeiro.
- Cardoso, D. D. O., Lima, P. M., De Gregorio, M., Gama, J., França, F. M. (2011). Clustering data streams with weightless neural networks. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 201-206.
- Cardoso, D. O. (2012). Uma arquitetura para agrupamento de dados em fluxo contínuo baseada em redes neurais sem pesos. *Dissertação de Mestrado*, Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.
- Cardoso, D. O., Gama, J., De Gregorio, M., França, F. M. G., Giordano, M., Lima, P. M. V. (2013). WIPS: the WiSARD Indoor Positioning System. *In proceedings, European Symposium on Artificial Neural Networks (ESANN 2013)*, Computational Intelligence and Machine Learning. Bruges, Belgium, 24-26.
- Chiang, L. H.; Russel, E. L.; Braatz, R. D. (2001). Fault detection and diagnosis in industrial systems. *1ª Ed. London, UK*, Springer Vela.
- Clarkson, T. G., Ng., C. K., Gorse, D., Taylor, J. G. (1992). Learning probabilistic RAM nets using VLSI structures. *IEEE Transactions on Computers*, 41(12), 1552-1561.
- Conti, V., Militello, C., Sorbello, F., Vitabile, S. (2009). An embedded fingerprints classification system based on weightless neural networks. *In Proceedings of the conference on New Directions in Neural Networks: 18th Italian Workshop on Neural Networks: WIRN 2008*, 67-75.
- Cybenko, G. (1988). Continuous valued neural networks with two hidden layers are sufficient. *Technical report, Department of Computer Science*, Tufts University.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoid function. *Mathematics of Control, Signals and Systems*, 2, 303-314.
- D'Angelo, M. F., Palhares, R. M., Takahashi, R. H., Loschi, R. H., Baccharini, L. M., Caminhas, W. M. (2011). Incipient fault detection in induction machine stator-winding using a fuzzy-Bayesian change point detection approach. *Applied Soft Computing*, 11(1), 179-192.
- de Oliveira, W. R., Ludermir, T. B. (1992). Turing machine simulation by logical neural networks. *Artificial Neural Networks*, 2, 663-667.
- Downs, J. J., Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3), 245-255.
- Duda, R. O., Hart, P. E. (1973). Pattern classification and scene analysis. *Wiley*. New York.
- França, H. L., Silva, J. C. P., Lengerke, O., França, F. M. G., Dutra, M. S. (2009). Um sistema de visão artificial para o controle de perseguição de movimento por uma plataforma

- Stewart. *In Anais 2º Congresso Internacional de Ingeniería Mecatrónica Unab, Bucaramanga.*
- Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – A survey and some new results. *Automatica*, 26(3), 459–474.
- Gorse, D., Romano-Critchley, D. A., Taylor, J.G. (1997). A pulse-based reinforcement algorithm for learning continuous functions. *Neurocomputing*, 14(4), 319-344.
- Gorse, D., Taylor, J. G. (1988). On the equivalence and properties of noisy neural networks and probabilistic RAM nets. *Physics Letters*, 131(6), 326-332.
- Gorse, D., Taylor, J. G. (1991). A continuous input RAM-based stochastic neural model. *Neural Networks*, 4, 657-665.
- Grieco, B., Lima, P., De Gregorio, M., França, F. M. (2010). Producing pattern examples from mental images. *Neurocomputing*, 73(7), 1057-1064.
- Hagan, M., Menhaj, M. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989-993.
- Hayashi, M. Y. (2013). *Pré-sal: desafios e oportunidades (um sonho possível)*. Revista Interesse Nacional. Ano 6, 22.
- Hecht-Nielsen, R. (1990). *Neurocomputing*. USA-Addison-Wesley, Publishing Company.
- Isermann, R. (2006). *Fault-Diagnosis Systems: An introduction from fault detection to fault tolerance*. Springer Verlag.
- Isermann, R., Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5), 709–719.
- Johannesmeyer, M. C (1999). Abnormal situation analysis using pattern recognition techniques and historical data. *M.Sc. Thesis, University of California, Santa Barbara, CA.*
- Kan, W. K., Aleksander, I. (1987). A probabilistic logic neuron network for associative learning. *In Proc. of the IEEE International Conference on Neural Networks*, 2, 541-548.
- Korbicz, J., Witeczak, M. (2005). Soft computing in fault detection and isolation. *In EUSFLAT Conference*, 2-3.
- Lemos, A. P. (2011). *Modelagem nebulosa evolutiva: Novas topologias e algoritmos de aprendizagem. Tese de Doutorado em Engenharia Elétrica*. Universidade Federal de Minas Gerais, Minas Gerais, Brasil.
- Lemos, A. P., Caminhas, W. M., Gomide, F. (2010). New uninorm-based neuron model and fuzzy neural networks. *In Proceedings of NAFIPS 2010 - North American Fuzzy Information Processing Society's 2010*, 1-6.
- Linden, R (2008). Algoritmos genéticos na indústria do petróleo: Uma visão geral. *Revista da Engenharia de Instalações no Mar*, 1.
- Ludermir, T. B., de Carvalho, A., Braga, A. P., De Souto, M. C. P. (1999). Weightless neural models: A review de current and past work. *Neural Computing Surveys*, 2, 41-61.
- Maia, R. D., Inácio, M. J., Lemos, A. P., Caminhas, W. M. (2009). Aprendizado participativo aplicado à detecção e diagnóstico on-line das falhas em sistemas dinâmicos. *In IX Congresso Brasileiro de Redes Neurais/Inteligência Computacional*, Minas Gerais, Brasil.
- McCulloch, W. S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-137.

- McElroy, B., Howells, G. (2011). Automated adaptation of input and output data for a weightless artificial neural network. *International Journal of Database Theory and Application*, 4(3), 49-58.
- Medeiros, J. P. (2009). Um estudo e implementação de algoritmos inteligentes para detecção e diagnóstico de falhas na medição de gás natural. *Dissertação de Mestrado, Programa de Pós-Graduação em Ciência e Engenharia do Petróleo da Universidade Federal do Rio Grande do Norte*, Rio Grande do Norte, Natal, Brasil.
- Mpofu, N. (2006). Forecasting Stock Prices Using a Weightless Neural Network. *Journal of Sustainable Development in Africa*, 8, 115-119.
- Myers, C., Aleksander, I. (1988). Learning algorithms for probabilistic logic nodes. *In Abstracts of I Annual INNS Meeting*, 205-205.
- Myers, C., Aleksander, I. (1989). Output functions for probabilistic logic nodes. *In Proceedings of the IEE International Conference Neural Networks*, 310-314.
- Nurmaini, S., Hashim, S. Z. M., Jawawi, D. N. A. (2009). Modular weightless neural network architecture for intelligent navigation. *International Journal of Advanced in Soft Computing and its Application*, 1(1), 1-18.
- Nurmaini, S., Tutuko, B. (2011). A new classification technique in mobile robot navigation. *In TELKOMNIKA*, 9(3), 453-464.
- Oliveira, J. C. M. (2002). Implementação de Autômatos Probabilísticos em Redes Neurais sem Pesos. *Dissertação de Mestrado*, Departamento de Informática. Universidade Federal de Pernambuco (UFPE), Pernambuco, Brasil.
- Oliveira, J. C. M., Pontes, K. V., Sartori, I., Embiruçu, M. (2017). Fault detection and diagnosis in dynamic systems using weightless neural networks. *Expert Systems with Applications*, 84, 200-219.
- Pearlmutter, B. (1992). Gradient descent: second order momentum and saturation error. *In J. E. Moody, S. Hanson, and R. Lippmann, editors, Advances in Neural Information Processing Systems*, 2, 887-894.
- Pietrosanto, A., Betta, G. (2000). Instrument fault detection and isolation. State of the art and new research trends. *IEEE Transactions on Instrumentation Measurement*, 49(1), 100-107.
- Riedmiller, M. (1994). Rprop - description and implementation details. *Technical report. University of Karlsruhe*.
- Rivera, M. H. M. (2007). Diagnóstico de falhas em sistemas a eventos discretos: Uma proposta de aplicação em processos de separação óleo-gás. *Dissertação de Mestrado*, Programa de Engenharia Elétrica. Universidade Federal do Rio de Janeiro.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386-408.
- Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986a). Learning internal representations by error propagation. *In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Eds.), Parallel Distributed Processing I*, Cambridge, MA: MIT Press, 318-362.
- Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986b). Learning representations by back-propagation of errors. *Nature*, 323, 533-536.



- Rumelhart, D. E., McClelland, J. L. (1986). *Parallel distributed processing*. Cambridge, MA, USA: MIT press, 1, 184-184.
- Sartori, I (2010). Gerenciamento de eventos anormais de uma unidade de processamento de gás natural através de sistemas de detecção e diagnóstico de falhas. *Exame de Qualificação ao Doutorado*, Programa de Engenharia Industrial. Universidade Federal da Bahia-UFBA, Salvador, Bahia, Brasil.
- Sartori, I. (2012). Uma abordagem de desenvolvimento de sistemas de detecção e diagnóstico de falhas baseada em estudos prévios de confiabilidade e sua aplicação ao processamento industrial de gás natural. *Tese de Doutorado*, Programa de Engenharia Industrial. Universidade Federal da Bahia-UFBA, Salvador, Bahia, Brasil.
- Sartori, I., Amaro, C. A., Souza Jr, M. B. D., Embiruçu, M. (2011). Detecção, diagnóstico e correção de falhas: Uma proposição consistente de definições e terminologias. *Ciência & Engenharia*, submetido.
- Silva, D. R. C. (2008). Sistema de detecção e isolamento de falhas em sistemas dinâmicos baseado em identificação paramétrica. *Tese de Doutorado*, Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Natal, Brasil.
- Singhal, A., Seborg, D. E. (2002). Pattern matching in multivariate time series databases using a moving-window approach. *Industrial & Engineering Chemistry Research*, 41(16), 3822-3838.
- Sirlantzis, K., Howells, G., Gherman, B. (2009). Novel Modular Weightless Neural Architectures for Biometrics-based Recognition. *European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning*, Bruges, Belgium.
- Soares, A. S., Galvão, R. K. H. (2010). Fault detection using linear discriminant analysis with selection of process variables and time lags. *International Conference on Industrial Technology, ICIT2010*, IEEE, 217-222.
- Sorsa, T., & Koivo, H. N. (1993). Application of artificial neural networks in process fault diagnosis. *Automatica*, 29(4), 843-849.
- Vaidyanathan, R., & Venkatasubramanian, V. (1992). Representing and diagnosing dynamic process data using neural networks. *Engineering Applications of Artificial Intelligence*, 5(1), 11-21.
- Vaz, c. E. M., Maia, j. L. P., Santos, W. G. Dos (2008). Tecnologia da indústria do gás natural. 1a ed., *Blucher*, São Paulo, Brasil.
- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., Yin, K. (2003). A review of process fault detection and diagnosis part I: Quantitative Model-Based Methods. *Computers & Chemical Engineering*, 27(3), 293-311.
- Vidal, F. S., de Castro Carneiro, H. C., Ferreira, P. F., Tibúrcio, P. G., de Janeiro, U. R. (2013). Identificação de emoções a partir de expressões faciais com redes neurais sem pesos. In *Simpósio Brasileiro de Automação Inteligente (SBAI)*, Ceará, Brasil.
- Yáñez, C., Camacho, O. (2005). Pattern recognition and classification using Weightless Neural Networks (WNN) and Steinbuch Lernmatrix. In *Optics and Photonics 2005 - International Society for Optics and Photonics*, 59160.
- Yao, Q., Beetner, D., Wunsch, D. C., Osterloh, B. (2003). A RAM-based neural network for collision avoidance in a mobile robot. In *Proceedings of the International Joint Conference on Neural Networks*, 4, 3157-3160.

## CHAPTER III

### Chapter 3: Faults Detection and Diagnosis in Dynamic Systems Using WiSARD Weightless Neural Networks

#### Abstract

This work examines Fault Detection and Diagnosis (FDD) based on Weightless Neural Networks (WNN) with applications in univariate and multivariate dynamic systems. WNN use neurons based on RAM (Random Access Memory) devices. These networks use fast and flexible learning algorithms, which provide accurate and consistent results, without the need for residual generation or network retraining, and therefore they have great potential use for pattern recognition and classification [Ludermir, *et al.*, (1999)]. The proposed system firstly executes the selection of attributes (in the multivariable case) and does the time series mapping of the data. In the intermediate stage, the WNN performs the detection and diagnosis per class. The network outputs are then passed through a clustering filter in the final stage of the system, if a diagnosis per fault groups is necessary. The system was tested with two case studies: one was an actual application for the temperature monitoring of a sales gas compressor in a “Natural Gas Processing Unit (NGPU)”; and the other one uses simulated data for an industrial plant, known in the literature as “Tennessee Eastman Process”. The results show the efficiency of the proposed systems for FDD with classification accuracies of up to 98.78% and 99.47% for the respective applications.

**Keywords:** Fault Detection and Diagnosis; Weightless Neural Networks; Time Series.

## CAPÍTULO III

### Capítulo 3: Detecção e Diagnóstico de Falhas em Sistemas Dinâmicos Usando Redes Neurais Sem Pesos WiSARD

#### Resumo

Este capítulo aborda a detecção e o diagnóstico de falhas baseados nas redes neurais sem pesos com aplicações em sistemas dinâmicos univariáveis e multivariáveis. As redes neurais sem pesos utilizam neurônios baseados em dispositivos de memórias RAM (*Random Access Memories*, memórias de acesso randômico ou aleatório). Essas redes apresentam algoritmos de aprendizagem rápidos e flexíveis, precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento das redes e um potencial elevado para o reconhecimento e classificação de padrões (Ludermir *et al.*, 1999). O sistema proposto executa, previamente, a seleção de atributos (multivariáveis) por classe e o mapeamento temporal dos dados e, na sua etapa intermediária a rede neural sem pesos realiza a detecção e o diagnóstico por classe. As saídas da rede são levadas a um filtro de agrupamento, na etapa final do sistema, se um diagnóstico por grupos de falhas for necessário. O sistema foi testado com dois estudos de casos. Um deles corresponde a uma aplicação real para monitoramento de temperatura de um compressor de gás para a venda instalado em uma “Unidade de Processamento de Gás Natural (UPGN)”. O outro utiliza dados simulados para uma planta industrial, conhecido na literatura como *Tennessee Eastman Process* (TEP). Os resultados mostram o sucesso dos sistemas para a detecção e o diagnóstico de falhas com percentuais de acertos na classificação de até 98,78% e 99,47% para as respectivas aplicações.

**Palavras-chave:** Detecção e Diagnóstico de Falhas; Redes Neurais sem Pesos; Séries Temporais.

### 3.1 Introdução

A crescente demanda por processos industriais mais produtivos e confiáveis, frente aos avanços tecnológicos, tem resultado em uma automação mais complexa e com maior disponibilidade de dados. Consequentemente, aumenta-se a necessidade de sistemas de supervisão e controle mais eficientes, com destaque para a detecção e o diagnóstico de falhas (FDD, *Fault Detection and Diagnosis*) em ambientes dinâmicos. Num sistema de produção, uma falha representa uma condição anormal de funcionamento, provocada por fatores como: erros de projeto; erros de instalação; uso indevido; ou efeitos de degradação natural. Dispor de mecanismos para a detecção antecipada e segura de falhas diminui os riscos de mau funcionamento ou paradas não programadas do sistema. Consequentemente, aumenta-se a confiabilidade nos equipamentos, evitam-se perdas materiais e, até mesmo, acidentes de trabalho e ambientais (Blázquez e Miguel, 2005; Burns, 2006; Chiang *et al.*, 2001; Fan *et al.*, 2013; Romano e Kinnaert, 2006; Yang e Liu, 1998).

Os sistemas de FDD, considerando o conhecimento *a priori* utilizado para o seu desenvolvimento, são desenvolvidos com base em “dados históricos” do processo ou “modelos”. Tais abordagens podem ser subdivididas em métodos “quantitativos” e “qualitativos” (Venkatasubramanian *et al.*, 2003a, b). Os sistemas baseados em “dados históricos” do processo conseguem extrair características relevantes dos dados, de modo a mapear as relações e os limites existentes entre as classes ou agrupamentos considerados. Neste grupo, alguns métodos “quantitativos” podem ser destacados: classificadores estatísticos (Luo *et al.*, 2011; Ma *et al.*, 2010; Soares e Galvão, 2010;); redes neurais (Lau *et al.*, 2013; Leite *et al.*, 2012; Sartori, 2012; Zarei, 2012); lógica *fuzzy* (Andrade *et al.*, 2011; Andrade, 2012; Lau *et al.*, 2013; Leite *et al.*, 2012; Li *et al.*, 2013; Silva *et al.*, 2012); análise de componentes principais (PCA, *Principal Components Analysis*) (Barragan *et al.*, 2016; Jiang *et al.*, 2013; Lau *et al.*, 2013); método dos mínimos quadrados parciais (Zhang *et al.*, 2010); transformadas de wavelet (Barragan *et al.*, 2016). Entre os métodos “qualitativos” destacam-se os sistemas especialistas (Saravanan *et al.*, 2009; Zadeh, 2008; Wang, 2012) e a análise qualitativa de tendências (Maurya *et al.*, 2007). Nos sistemas baseados em “modelos”, o comportamento real monitorado é comparado com a resposta obtida por um modelo representativo do processo. O resultado dessa comparação supervisionada é um vetor de resíduos usado para detectar a presença de falhas. Neste grupo é possível destacar os seguintes métodos “quantitativos”: observadores de estados e de saídas (Chetouani, 2008; Kalman, 1960); equações e espaço de paridade (Beckerle *et al.*, 2012; Blesa *et al.*, 2014; Zakharov, *et al.*, 2013; Zhong *et al.*, 2015); filtro de Kalman estendido (Kalman, 1960; Patwardhan e Shah, 2006); máquina de vetores de suporte (Deng *et al.*, 2011; Duan *et al.*, 2016; Park *et al.*, 2011); e os métodos de identificação e estimação de parâmetros (Johansson *et al.*, 2006; Pouliezos *et al.*, 1989). Numa abordagem “qualitativa” destacam-se: árvores de falha (Nguyen e Lee, 2008; Simões Filho, 2006); simulação qualitativa (Berleant, 1991); teoria qualitativa de processo (Venkatasubramanian *et al.*, 2003a); e grafos direcionados com sinais, a exemplo das redes Bayesianas (Ji e Meng, 2015; Luo *et al.*, 2012; Xiao *et al.*, 2014). Serdio *et al.* (2014) desenvolveram abordagens que combinam dados históricos, extração automática de modelo e geração residual, que parecem úteis para detecção e isolamento de falhas com falhas abruptas e incipientes.

Muitas pesquisas têm mostrado um diferencial positivo e crescente das técnicas de base estatística e da inteligência artificial para a solução de problemas de FDD, destacando-se os sistemas híbridos envolvendo lógica *fuzzy*, algoritmos genéticos e principalmente as redes neurais (Herzog *et al.*, 2009; Hu e Yi, 2016; Liao *et al.*, 2012; Niaki e Abbasi, 2005; Rajpal *et al.*, 2006; Santosh *et al.*, 2007; Sartori *et al.*, 2012; Sharma *et al.*, 2015). A tolerância a ruídos apresentada pelas redes neurais artificiais (Özyurt e Kandel, 1996) e as habilidades em lidar

com imprecisões, ambiguidades e incertezas nas informações apresentadas pela lógica *fuzzy* (Leite *et al.*, 2012), tornam os sistemas híbridos “neuro-*fuzzy*” bastante utilizados (Hell *et al.*, 2008). Essas técnicas se adequam bem em sistemas não lineares, por não requerem modelos matemáticos explícitos (Angelov e Yager, 2012; Barty *et al.*, 2006; Bocaniala e da Costa, 2006; Li *et al.*, 2013; Lo *et al.*, 2009; Ma *et al.*, 2010; Rigatos e Zhang, 2009). De acordo com Sartori *et al.* (2016), a maioria das aplicações de sistemas de FDD está voltada para unidades de geração e distribuição de energia, equipamentos da indústria de processos (reatores, colunas, sensores e atuadores), motores e rolamentos. O referido estudo aponta as técnicas de redes neurais, lógica *fuzzy*, análise de componentes principais, filtro de Kalman, máquina de vetores de suporte, algoritmo genético e sistemas especialistas, em ordem decrescente, como as mais utilizadas.

De modo geral, as abordagens citadas se adaptam bem na detecção e no diagnóstico de falhas abruptas. No contexto de falhas incipientes em sistemas dinâmicos, a solução do problema de FDD é complexa e os métodos encontrados na referida literatura ainda deixam lacunas, dentre as quais se destacam: problemas multivariáveis e com diversidade no número de classes consideradas; processos não lineares com falhas incipientes presentes em duas ou mais das classes consideradas; e dificuldade na obtenção de dados históricos com aplicações reais. Dentro desse contexto, este trabalho tem como principal contribuição a proposição de um sistema de detecção e diagnóstico de falhas incipientes em sistema dinâmicos, sem a necessidade de modelo matemático ou cálculo de resíduos, e com uma baixa taxa de alarmes falsos. O sistema proposto tem como base as Redes Neurais Sem Pesos (RNSP), propostas inicialmente por Aleksander (1967). As RNSP são modelos digitais baseados em dispositivos de memória de acesso randômico (RAM, *Random Access Memories*). Diferente dos modelos neurais convencionais, o aprendizado acontece em “memórias” inseridas no próprio neurônio, em forma de tabelas-verdade. A diversidade de uso dessas memórias traz vantagens às RNSP em relação aos modelos com pesos, quais sejam: semelhança com os sistemas digitais convencionais; algoritmos de aprendizagem rápidos e flexíveis; precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento das redes; e principalmente o grande potencial das RNSP para o reconhecimento e a classificação de padrões.

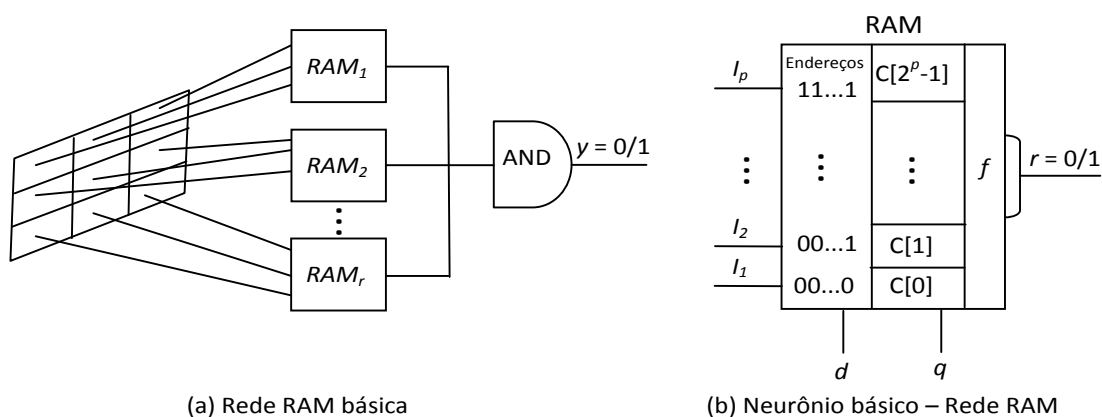
As habilidades das RNSP para os problemas de reconhecimento e classificação de padrões podem ser comprovadas em diversas áreas, dentre as quais se destacam: “reconhecimento de dígitos e impressões digitais” (Bandeira *et al.*, 2009; Conti, *et al.*, 2009; Grieco *et al.*, 2010); “reconhecimento de faces e impressões faciais” (Araújo, 2011; Sirlantzis *et al.*, 2009; Subhashini e Nagarajan, 2014); “navegação de robôs” (McElroy e Howells, 2011; Nurmaini *et al.*, 2009); “agrupamento de fluxos de dados” (Cardoso *et al.*, 2011); e “previsão de séries temporais” (De Souza *et al.*, 2010; Mpofu, 2006). Porém, nenhum trabalho que trate os problemas de FDD com o uso das RNSP é encontrado na literatura. Os trabalhos apresentados por De Gregorio e Giordano (2014) e Cardoso *et al.*, (2012) abordam este contexto, mas não lidam com problemas de FDD. De Gregorio e Giordano (2014) usaram as RNSP para o problema de detectar mudanças no campo de visão de uma câmera. O sistema proposto, chamado CwisarDH, usa um discriminador para cada ponto de cobertura do vídeo com o conceito de cor. Cardoso *et al.*, (2012) apresentaram o sistema StreamWiSARD para o agrupamento de dados de fluxo com janela deslizante. O sistema consiste em discriminadores WiSARD como unidades primárias e é capaz de definir *clusters* de alta qualidade, restrito a um pequeno número de *microclusters*. A ausência de sistemas de FDD baseados nas RNSP, e o potencial dessas redes para o reconhecimento e a classificação de padrões, foram os principais fatores que motivaram o desenvolvimento deste trabalho.

Para validar o sistema desenvolvido foram considerados dois estudos de caso. O primeiro compreende uma base de dados univariáveis para os sensores de temperatura de um compressor de gás para a venda instalado em uma Unidade de Processamento de Gás Natural (UPGN) da Petrobras, localizada no município de Pojuca-BA, Brasil (Andrade, 2012; Andrade *et al.*, 2011; Sartori, 2012). No segundo estudo de caso foram utilizados dados multivariáveis de um simulador de uma planta industrial considerada como um *benchmark* (referência) na área de detecção e diagnóstico de falhas, conhecida como *Tennessee Eastman Process* (TEP) (Downs e Vogel, 1993; Ricker, 1995). Na aplicação multivariável foi utilizado um algoritmo de seleção de atributos por classe, proposto por Vale *et al.* (2010) e com modificações propostas neste trabalho.

Na continuidade deste capítulo, a seção 3.2 apresenta os fundamentos sobre redes neurais sem pesos. Na seção 3.3 são apresentados o sistema FDD-WiSARD (FDD-WiSARD, *detecção e diagnóstico de falhas com base no dispositivo WiSARD*) e a metodologia utilizada para os treinamentos e testes realizados. Na seção 3.4 são apresentados os dois estudos de caso utilizados com os seus respectivos resultados. Por último, na seção 3.5, são apresentadas as conclusões sobre os sistemas FDD-WiSARD propostos para os estudos casos considerados.

### 3.2 Redes Neurais sem Pesos

Aleksander (1967), um dos pioneiros na área de RNA (Redes Neurais Artificiais), propôs um modelo de neurônio/ nodo totalmente digital, baseado em dispositivos de memória de acesso randômico RAM. O modelo deu origem a uma classe de neurônios baseados em lógica Booleana, também conhecidos como neurônios sem pesos (*weightless neurons*) ou neurônios baseados em RAM (*RAM-based neurons*). As redes neurais compostas por esses neurônios são denominadas de Redes Neurais Sem Pesos (RNSP). O aprendizado nessas redes consiste em mudanças nos conteúdos de memórias em tabelas-verdade inseridas nas RAM. Um nodo sem pesos aprende ou reconhece apenas parte (subpadrão) do padrão de entrada apresentado à rede. Os subpadrões colocados como entradas para as RAM correspondem aos endereços acessados e são definidos por algum critério de mapeamento do padrão de entrada (atributo qualitativo ou quantitativo) binarizado para a rede, a exemplo do mostrado na Figura 3.2. Por isso, a generalização nas RNSP com nodos RAM só acontece em nível da rede (Ludermir, *et al.*, 1999). A estrutura da rede mostrada na Figura 3.1(a) refere-se ao modelo neural sem pesos básico, denominado de rede RAM, totalmente binário e definido por Aleksander (1967), cujo neurônio é detalhado na Figura 3.1(b). Nas RNSP somente as etapas de treinamento e teste são executadas. O teste é realizado assim que o treinamento é finalizado e em muitos trabalhos é referenciado como validação.



**Figura 3.1:** Estrutura da Rede e do Neurônio RAM [Oliveira *et al.*, 2017].

Um nodo RAM com  $p$  entradas tem  $2^p$  posições de memória, endereçadas pelos  $p$  bits do vetor  $w = \{w_1, w_2, \dots, w_p\}$ . Um sinal binário  $I = \{I_1, I_2, \dots, I_p\}$  nas linhas de entrada acessa apenas uma dessas locações, aquela para a qual  $w = I$ . O terminal  $q$ , na Figura 3.1(b), define a fase em que o nodo se encontra: aprendizagem; ou teste. Se a rede estiver em processo de aprendizado o valor de  $q$  é igual a 1 e o conteúdo acessado pelas  $p$  linhas de entrada é modificado para o valor desejado (0 ou 1), fornecido pelo terminal  $d$ . O conteúdo  $C[a]$  em todas as locações de memórias, sendo  $a$  o endereço acessado pelo padrão  $I$ , é inicializado com o valor 0 (zero). Na fase de teste, o bit armazenado na posição de memória  $C[a]$  é a saída da RAM  $r = C[a]$ , definida pela função identidade  $f$ . Se o valor encontrado na posição de memória  $C[a]$  for 0, significa que o último aprendizado ocorrido foi 0 ou que aquele endereço não foi acessado durante o treinamento. Assim como nas RAM, a saída  $y$  da rede RAM também é 0 ou 1. A saída da rede é igual a 1 somente se as saídas de todas as RAM forem iguais a 1. Caso contrário, o padrão será incluído na classe 0 como padrões não pertencentes à referida aplicação. Se, no problema em estudo, mais de duas categorias são requeridas, utiliza-se a estrutura “discriminador”, mostrada na Figura 3.2 e descrita na seção seguinte.

### 3.2.1 A Rede WiSARD

Uma rede composta por neurônios RAM pode ter vários tipos de arquiteturas, embora a mais utilizada seja a WiSARD (Wilkie, Stonham e Aleksander’s Recognition Device, dispositivo de reconhecimento de Wilkie, Stonham e Aleksander) proposta por Aleksander *et al.* (1984). A rede WiSARD é composta por dois ou mais discriminadores (mostrado na Figura 3.2 (Adaptação de Ludermir *et al.*, (2008))). Cada discriminador é treinado individualmente para representar uma única classe de padrões. A quantidade de RAM em cada discriminador depende do modelo de mapeamento dos padrões de entrada para a rede, como exemplo, o apresentado na Figura 3.2. Cada subpadrão aprendido ou testado por uma RAM é representado por uma  $n$ -tuple (Bledsoe e Browning, 1959), conforme modelo de mapeamento usado. Na fase de teste, a mesma entrada é apresentada a todos os discriminadores. Assim, cada RAM identifica o endereço de memória apontado pela  $n$ -tuple correspondente e apresenta como saída o valor armazenado, 0 ou 1, como indicados na Figura 3.2. A saída de cada discriminador corresponde ao número de RAM com saída igual a 1, somatório de valores 1, e o padrão de teste é reconhecido pelo discriminador com maior somatório.

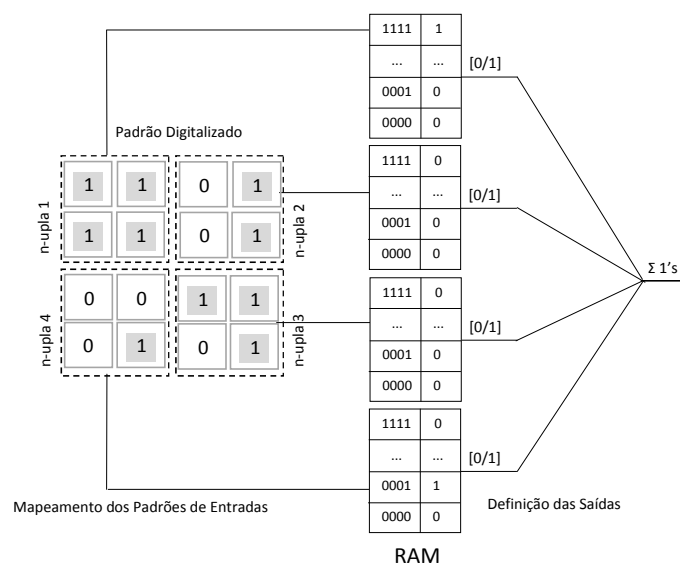
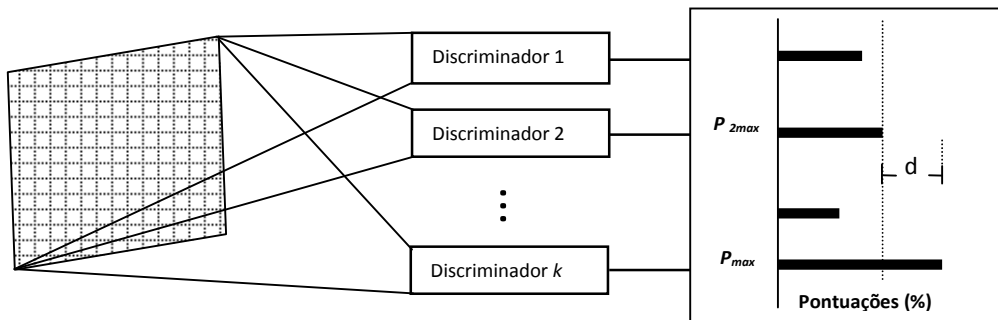


Figura 3.2: Representação Esquemática de um Discriminador.

Além da pontuação calculada para cada discriminador, a rede WiSARD fornece um nível de confiança ( $C_w$ ) para o reconhecimento realizado, que é calculada por:

$$C_w = \frac{P_{max} - P_{2,max}}{P_{max}} \quad (3.1)$$

onde  $P_{max}$  é a pontuação máxima efetivamente obtida por um dos discriminadores e  $P_{2,max}$  é a segunda maior pontuação. A Figura 3.3 exibe o resultado do reconhecimento por uma WiSARD com  $k$  discriminadores. Nesta ilustração o discriminador  $k$  apresentou a pontuação máxima e o discriminador 2 a segunda maior pontuação.



**Figura 3.3:** WiSARD - Reconhecimento com  $k$  Discriminadores [Adaptado de Aleksander, (2009)].

Quando dois ou mais discriminadores apresentam a maior pontuação, a confiança é nula. Neste caso, se o empate acontece com a pontuação máxima possível, usa-se o critério de aleatoriedade para definir um dos discriminadores empatados como vencedor. Porém, o uso da aleatoriedade diminui a acurácia da rede, pois o padrão testado tem grande possibilidade de ser classificado erroneamente. Outra dificuldade apresentada pela rede WiSARD padrão é a saturação dos endereços de memória. A saturação acontece quando todos ou quase todos os endereços de memória das RAM são utilizados. Além disso, a rede WiSARD original foi proposta inicialmente para o reconhecimento e classificação de imagens digitais, baseada em um modelo totalmente binário. Esta dificuldade, no entanto, tem sido superada, podendo ser encontrados alguns trabalhos na literatura que adaptaram a WiSARD padrão para aplicações e procedimentos com valores reais, a exemplo dos trabalhos apresentados por Araújo (2011), Cardoso *et al.* (2011), De Souza *et al.* (2010), Souza *et al.* (2012) e Subhashini e Nagarajan (2014). Técnicas também têm sido propostas para superar as limitações da aleatoriedade e da saturação dos endereços, o que será discutido na próxima seção.

### 3.2.2 A Técnica de *Bleaching*

A técnica de “*bleaching*” (refinamento) consiste em usar um valor limiar para amenizar o uso do critério de aleatoriedade e realizar um desempate mais confiável entre os discriminadores de uma rede WiSARD (Carvalho *et al.*, 2013). Com esse mecanismo, os endereços de memória nas RAM da WiSARD passam a armazenar valores inteiros, ao invés de valores 0 e 1. O valor inteiro armazenado em um endereço de memória representa o número de vezes em que aquele endereço foi acessado durante o treinamento. Na fase de teste, a saída de uma RAM é igual a 1 se o valor inteiro armazenado for igual ou superior ao valor do limiar. Assim, o valor do limiar pode ser definido de duas formas (Carvalho *et al.*, 2013): em sua forma “convencional” ou em uma versão “incremental” (ou porcentual).

No *bleaching* convencional, o mesmo valor de limiar é usado para todos os discriminadores. Como o limiar é inicializado com o valor zero, a verificação inicial de pontuação das RAM se



comporta como se a técnica não fosse utilizada, ou seja, se o conteúdo acessado é maior do que 0, a saída da RAM é 1. Porém, se dois ou mais discriminadores apresentarem o mesmo somatório de valores 1, o valor do limiar é acrescido de uma unidade. Com isso, os conteúdos das RAM serão novamente verificados de modo a reduzir o número de RAM com saída igual a 1. Este procedimento é repetido até que um discriminador seja eleito ou até que todos parem de pontuar, situação em que a escolha é feita de modo aleatório.

No *bleaching* percentual, o valor do limiar varia de acordo com a quantidade de aprendizado em cada discriminador. Dessa forma, evita-se que uma categoria com mais treinamentos seja privilegiada, pois os limiares percentuais têm a mesma representatividade e criam um equilíbrio entre os discriminadores. Assim, é possível determinar o grau de importância de um padrão testado. Para isso, utiliza-se o *bleaching* convencional e o número de treinamentos realizados por cada discriminador. Neste caso, a saída  $y$  de um RAM  $r$  é obtida com a Equação 3.2:

$$y(r) = \begin{cases} 1 & \text{se } \frac{C[a]}{nt(D_i)} > \frac{\text{Bleaching convencional}(\ )}{nt(D_{hf})} \\ 0 & \text{caso contrário} \end{cases} \quad (3.2)$$

onde  $C[a]$  é o conteúdo do endereço  $a$  acessado pelo padrão de entrada  $I$ ,  $nt(D_i)$  é o número de treinamentos do discriminador  $i$  contendo  $C[a]$ , com  $1 \leq i \leq k$  ( $k$  é quantidade de discriminadores com  $r$  RAM cada),  $i = 0, 1, 2, \dots$ , são os possíveis valores para o *bleaching* convencional e  $nt(D_{hf})$  é o número de treinamentos realizado pelo discriminador referente à classe com maior número de padrões.

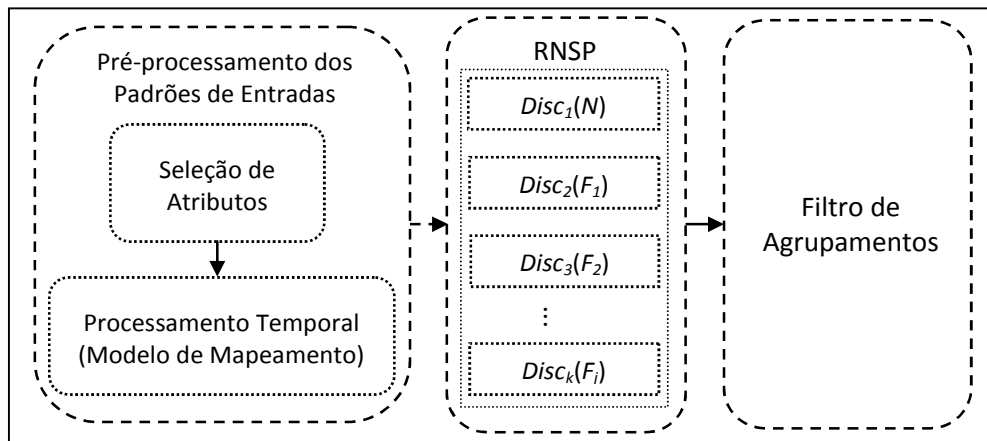
A técnica do *bleaching* pode ser utilizada também para amenizar a “saturação” dos endereços de memória, também mencionada na seção anterior. Para isso, toma-se como referência um valor de limiar para apagar gradativamente os conteúdos dos endereços de memória menos utilizados. Os endereços apagados são relativos a padrões com pouca representatividade ou não pertencentes à classe aprendida por um referido discriminador. Nesse caso, o valor de limiar representa a frequência mínima de escrita para que um conteúdo de memória não seja apagado. A implementação do procedimento consiste em armazenar, nos endereços de memória das RAM, o conteúdo e o último instante de acesso. Assim, quando um endereço de memória fica fora do limiar temporal considerado, o seu conteúdo é apagado. Se todas as posições de todas as RAM de um discriminador forem apagadas, esse discriminador poderá ser descartado por não armazenar qualquer informação. Por isso, o *bleaching* temporal é mais adequado para algoritmos não supervisionados, nos quais a quantidade de discriminadores é definida durante o processo de aprendizagem (Cardoso *et al.*, 2011; Souza *et al.*, 2012).

### 3.3 Sistema de Detecção e Diagnóstico de Falhas FDD-WiSARD Proposto

#### 3.3.1 Estrutura do Sistema

O sistema proposto neste trabalho realiza a detecção de falhas como um problema de classificação, no qual um padrão de entrada  $x = [a_1, a_2, \dots, a_n]$ , definido como vetor  $n$ -dimensional e  $a_i$  as variáveis (atributos) medidas, é considerado como estado de “operação normal” ou de “falha”. O diagnóstico, por outro lado, consiste na partição do estado de falha em subclasses com rótulos definidos pelos diferentes tipos de falhas considerados. Para isso, as variáveis (atributos)  $a_1, a_2, \dots, a_n$  devem conter informações que possibilitem a classificação desejada (Soares e Galvão, 2010). Assim, dado um padrão de entrada  $x$  e uma variável de resposta  $y$ , busca-se uma aproximação para a relação  $y = f(x)$ . Nessa relação  $y$  é um rótulo de classe com valor no conjunto  $\{C_1, C_2, \dots, C_k\}$ , com  $k \in \mathbb{N}$  e a função  $f(x)$  define, de forma

aproximada, os limites para as classes consideradas. Num contexto temporal,  $x_t$  é o valor de entrada  $x$  no instante  $t$ , com  $a_{1,t}, a_{2,t}, \dots, a_{n,t}$  sendo os atributos utilizados para classificar o padrão  $x_t = [a_{1,t}, a_{2,t}, \dots, a_{n,t}]$  como estado de operação “normal” ou estado de “falha” no instante  $t$ . Em termos estruturais, o sistema FDD-WiSARD contém três etapas distintas e complementares, denominadas de (Figura 3.4): “pré-processamento dos padrões de entrada”; “rede neural sem pesos”; e “filtro de agrupamentos”.



**Figura 3.4:** Estrutura do Sistema Neural sem Pesos: FDD-WiSARD.

No referido sistema,  $Disc_k$  são os discriminadores com  $k$  igual ao número de classes,  $N$  é classe de padrões com operação Normal e  $F_j$  são classes de padrões com falhas. A “etapa 1” executa o pré-processamento dos padrões de entrada (vetores  $n$ -dimensional). Cada dimensão corresponde a uma variável (atributo) medida por um sensor ou obtida por um simulador. Para aplicações multivariáveis é usado o algoritmo de seleção de atributos por classe, proposto por Vale *et al.* (2010), com um passo adicional em sua versão original, conforme apresentado na seção 3.3.2.1. Também na etapa inicial é realizado o processamento temporal, conforme modelos de mapeamento descritos na seção 3.3.2.2. A “etapa 2” do sistema corresponde a uma RNSP do tipo WiSARD, composta por uma camada com  $k$  discriminadores treinados individualmente para representar uma única classe de padrões (Aleksander, 1967). Por último, a “etapa 3” do sistema corresponde a um filtro de agrupamento que reduz a dimensionalidade de saída da rede de  $k$  possibilidades para uma classificação final desejada. Essas etapas do sistema FDD-WiSARD são descritas com mais detalhes nas seções 3.3.2 a 3.3.4.

### 3.3.2 Pré-Processamento dos Padrões de Entrada

#### 3.3.2.1 Seleção de Atributos

A eficácia em um sistema de classificação de padrões consiste, em termos ideais, em realizar a partição ideal do espaço de estados para o problema avaliado. Porém, essa partição ideal, principalmente em sistemas dinâmicos, dificilmente será obtida. No entanto, uma boa solução perpassa por resultados que se aproximem da situação ideal, considerada como a melhor partição possível do espaço de estados. Para isso, é importante que os classificadores envolvidos sejam “distintos” nas generalizações de modo a tornar o conjunto de classificadores num sistema “preciso” e “eficaz” no reconhecimento e classificação de padrões. Assim, uma das formas de obter uma distinção mais precisa em sistemas com multi classificadores, para problemas multivariáveis, é a seleção de atributos (Kuncheva e Whitaker, 2003). O uso de diferentes subconjuntos de atributos para os classificadores envolvidos possibilita a distinção desejada nas generalizações. Além disso, espera-se que os erros de classificação aconteçam

com base nos limites estabelecidos para as diferentes classes, conforme distribuição do espaço amostral (Tsymbal *et al.*, 2003).

Neste trabalho foi usado o algoritmo de seleção prévia de atributos por classe denominado de RecPun (**Re**compensa/**Pu**nição). Esse algoritmo foi proposto por Vale *et al.* (2010) e se baseia na recompensa e punição para ordenar as importâncias dos atributos para as classes existentes. A ideia do modelo é que exista, no mínimo, um classificador por classe e que os atributos selecionados para uma dada classe sejam importantes somente para a referida classe.

O uso do algoritmo RecPun na rede WiSARD foi possível porque os discriminadores são treinados individualmente para representar uma única classe de padrões. Assim, para cada padrão processado, o algoritmo RecPun seleciona, de forma não excludente, uma quantidade fixa ( $N$ ) de atributos por classe. Para isso são utilizados dois processos de ordenação das importâncias dos atributos. O primeiro utiliza a variância ou a correlação de Spearman (Guilford, 1950) para a ordenação e seleção dos atributos por classe. O segundo calcula a recompensa e punição dos atributos por classe com base na ordenação realizada pelo primeiro processo, aqui referenciado pela variância. A segunda ordenação dos atributos é obtida por meio do parâmetro  $RP$  (**Re**compensa/**Pu**nição) obtido pelas Equações 3.3 a 3.5 (Vale *et al.*, 2010):

$$RP_i = Rec_i - Pun_i \quad (3.3)$$

$$Rec_i = V_{i,c} + \frac{NA}{NA + R_{i,c}} \quad (3.4)$$

$$Pun_i = \frac{1}{NC - 1} \cdot \sum_{c=1}^{NC, c \neq i} \left( V_{i,c} + \frac{NA}{NA + R_{i,c}} \right) \quad (3.5)$$

onde  $NC$  é o número de classes,  $V_{i,c}$  é o valor que representa a importância do atributo  $i$  para a classe  $c$ ,  $R_{i,c}$  é a posição do atributo  $i$  na classe  $c$ , e  $NA$  é o número total de atributos usados no conjunto de dados (Vale *et al.*, 2010).

Ao aplicar o algoritmo RecPun no estudo de caso 2 (dois), foram selecionados os mesmos atributos para a maioria das classes. Com o propósito de promover uma maior diferenciação entre os atributos selecionados para as classes consideradas, propõe-se aqui um passo adicional ao algoritmo RecPun, de modo que a relação biunívoca entre as classes permita no máximo 50% de atributos comuns. Para isso, os atributos de uma classe  $i$ , com  $i = 1, 2, \dots, k$ , são comparados com os atributos das classes anteriores. Nessa comparação biunívoca, se a quantidade de atributos comuns for superior a 50%, mantêm-se os primeiros 50% e os demais são substituídos por atributos de menor importância, obedecendo ao segundo processo de ordenação definido pelo algoritmo RecPun.

### 3.3.2.2 Mapeamento dos Padrões de Entrada

Os modelos de mapeamento descritos nesta seção têm a função de transformar os padrões de entrada em endereços de acesso às RAM e são referidos como: “mapeamento simples”; ou “mapeamento temporal”. No “mapeamento simples” (0), definido a seguir, os padrões de entrada são processados individualmente e apresentados à rede. No “mapeamento temporal” [(1) a (6)], definido a seguir, os padrões de entrada são processados em séries temporais com horizonte  $h > 1$ , com base em “média móvel” e “janela deslizante” (Hamilton, 1994). O mapeamento temporal transforma uma série temporal em um padrão comportamental para alimentar a RNSP, mais especificamente os neurônios (RAM) do discriminador representativo da classe do padrão comportamental obtido, quando da aplicação univariável. De modo comparativo, nos modelos neurais com pesos as entradas são ponderadas com os pesos e o

resultado dessa ponderação alimenta a função de ativação usada nos neurônios da rede (Herzog *et al.*, 2009; Liao *et al.*, 2012; Niaki e Abbasi, 2005; Rajpal *et al.*, 2006; Sartori, 2012; Santosh *et al.*, 2007). Em aplicações multivariáveis, o mapeamento temporal é realizado separadamente para cada atributo do padrão de entrada  $x_t$ , com base em um dos modelos de mapeamento definidos a seguir [(1) a (6)]. Neste caso, o padrão comportamental é a concatenação dos atributos médios encontrados e o treinamento acontece no discriminador representante da classe do padrão de entrada atual,  $x_t$ .

0) Mapeamento simples. Neste modelo, o padrão de entrada atual é processado individualmente e apresentado à rede no discriminador representante da classe do referido padrão.

1) Média móvel ponderada pelos desvios em distâncias de Hamming e atributo médio como representante da série temporal processada. Neste caso, a ponderação é feita pelos desvios, em distâncias de Hamming (Hamming, 1950), dos atributos nos tempos anteriores em relação ao atributo do padrão atual,  $x_t$ . A distância de Hamming entre duas sequências de mesmo comprimento é o número de posições nas quais elas diferem entre si, isto é, corresponde ao número de *bits* diferentes entre dois vetores de *bits* de igual tamanho. Para as distâncias de Hamming, os atributos da série são representados por 8 ou 32 *bits*. O atributo médio  $\bar{a}_i$  é obtido pela Equação 3.6:

$$\bar{a}_i = \frac{\sum_{t=1}^h a_{i,t} \cdot d_{a_{i,t}}}{\sum_{t=1}^h d_{a_{i,t}}} \quad (3.6)$$

onde  $d_{a_{i,t}} = |a_{i,t} - a_{i,h}|$  é o desvio do atributo  $a_{i,t}$  em relação ao padrão de entrada atual  $x_t$ , com  $i = 1, 2, \dots, n$  e  $t = 1, 2, \dots, h$ , e  $h$  é o horizonte da janela temporal processada.

2) Média móvel ponderada pelos desvios em distâncias de Hamming e atributo mais próximo do atributo médio como representante da série processada. Este modelo difere do modelo “1” ao eleger  $\bar{a}_i$  como sendo o atributo da série processada com menor desvio em relação ao atributo médio encontrado.

3) Média móvel ponderada pelos graus de pertinência e atributo médio como representante da série processada. Neste caso, o atributo médio é a média ponderada pelos graus de pertinência obtidos pela função de distribuição de probabilidade Gaussiana, definida pela Equação 3.7:

$$f(a_t, \sigma, \mu) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma^2}} \cdot e^{-\frac{1}{2} \left( \frac{a_t - \mu}{\sigma} \right)^2} \quad (3.7)$$

onde  $a_t$  é o atributo de entrada e  $\mu$  e  $\sigma$  são a média aritmética e o desvio padrão dos  $t$  valores da série temporal processada e  $\pi = 3,141592\dots$ . O atributo médio  $\bar{a}_i$  é obtido pela Equação 3.8:

$$\bar{a}_i = \frac{\sum_{t=1}^h a_{i,t} \cdot \Delta_{a_{i,t}}}{\sum_{t=1}^h \Delta_{a_{i,t}}} \quad (3.8)$$

onde  $\Delta_{a_{i,t}}$  é o grau de pertinência do atributo  $i$  para o padrão de entrada  $x_t$ , com  $i = 1, 2, \dots, n$  e  $t = 1, 2, \dots, h$ , e calculado conforme Equação 3.7.

4) Média móvel ponderada pelos graus de pertinência e atributo mais próximo do atributo médio como representante da série processada. Este modelo difere do modelo “3” ao eleger  $\bar{a}_i$  como sendo o atributo da série processada com menor desvio em relação ao atributo médio encontrado.

5) Média móvel pela Suavização Exponencial Simples (SES) e atributo médio como representante da série processada. Neste modelo, os pesos são ajustados pela Equação 3.9:

$$E_h = W \cdot a_h + (1 - W) \cdot E_{h-1} \quad (3.9)$$

onde  $h$  é o horizonte da janela temporal processada,  $a_h$  é o atributo da série original no período  $h$ ,  $E_h$  é o valor da série ajustada no período  $h$ ,  $E_{h-1}$  é o valor da série ajustada no período  $h - 1$  e  $W$  é o coeficiente de ajuste, pertencente ao intervalo  $0 < W < 1$ . A escolha de  $W$  é subjetiva: se o interesse está na tendência da série, se recomendam valores próximos de 0 (zero); se o interesse está na previsão da série, se recomendam valores próximos de 1.

6) Média móvel pela SES e atributo mais próximo do valor estimado como representante da série processada. Este modelo difere do modelo “5” ao eleger  $\bar{a}_i$  como sendo o atributo da série processada com menor desvio em relação ao valor estimado.

No mapeamento de séries temporais, os 6 modelos mencionados levam em consideração a média móvel dos atributos da janela temporal processada. A diferença entre eles é a forma como essa média é calculada. Selecionar um modelo de mapeamento adequado e seus parâmetros para uma aplicação específica é uma tarefa importante. Diferentes sistemas requerem características diferentes e pré-processamento diferente, que na abordagem usada podem ser alcançados com a seleção do modelo de pré-processamento mais apropriado entre os descritos acima e “ajustes” dos parâmetros relacionados.

### 3.3.3 Estrutura da Rede Neural sem Pesos

A Rede Neural sem Pesos consiste de uma “única” camada com  $k$  discriminadores. Os estados de operação Normal ( $N$ ) e de Falhas ( $F_1, F_2, \dots, F_j$ ) são representados pelos discriminadores  $D_N, D_{F1}, D_{F2}, \dots, D_{Fj}$ , respectivamente. Conforme descrito no início da seção 3.2, a saída de uma rede RAM “básica” (Figura 3.1 (a)) é igual a 1 (classe 1) somente se as saídas de “todas” as RAM forem iguais a 1. Caso contrário, o padrão será incluído numa outra classe (0) como padrões não pertencentes à referida aplicação. Assim, é correto afirmar que uma rede RAM básica reconhece somente padrões vistos no treinamento. Porém, como nas aplicações em estudo, mais de duas categorias são requeridas, utilizou-se a estrutura da rede WiSARD, descrita na seção 3.2.1. Nesta rede, cada discriminador corresponde a uma rede RAM “básica”, treinada com padrões de uma única classe. No entanto, a validação na rede WiSARD, conforme descrita nas seções 3.2.1 e 3.2.2, classifica o padrão de teste como da classe do discriminador com “maior” número de RAM com saída igual a 1. O critério do “maior” e não “todas” como na RAM básica, permite os discriminadores da rede WiSARD reconhecerem padrões não vistos durante o treinamento. Então, enquanto o método precisa de padrões de dados defeituosos, possui alguma habilidade para discriminar padrões diferentes dos apresentados no processo de treinamento, especialmente se os novos padrões forem semelhantes aos conhecidos. Embora não seja realizado aqui, sistemas de treinamento sem dados de um tipo específico de falha e testá-lo com dados contendo esse tipo de falha seria uma possível medida de robustez do sistema em relação a padrões de falha nunca antes vistos na fase de treinamento.

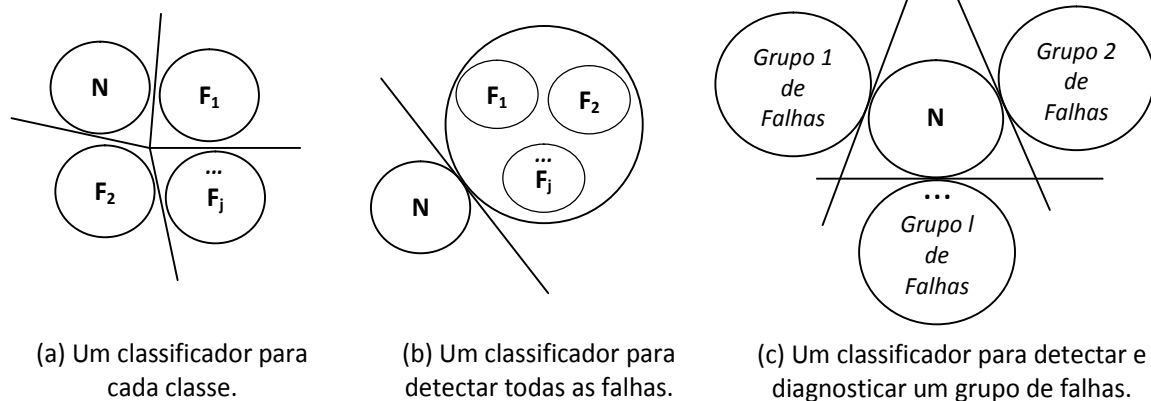
Como mostrado mais adiante na Figura 3.6, foram utilizados dois modelos (Padrão e Contador) de rede WiSARD para a tarefa de detecção e diagnóstico de falhas pelo sistema FDD-WiSARD,

descritos a seguir. A rede WiSARD Padrão (WP), conforme descrita na seção 3.2.1, possui uma única camada com dois ou mais discriminadores, treinados individualmente para representar uma única classe de padrões. Na fase de teste, a mesma entrada é apresentada a todos os discriminadores. Assim, cada nodo RAM identifica o endereço de memória apontado pela  $n$ -upla correspondente e retorna o valor armazenado 0 ou 1. A saída da rede corresponde ao discriminador com maior somatório de valores 1 retornados pelos nodos RAM. No caso de empate, usa-se o critério da aleatoriedade para definir o discriminador vencedor. A rede WiSARD Contador (WC) possui a mesma estrutura da WiSARD Padrão, com diferenças na funcionalidade. O “valor escalar” armazenado como conteúdo de memórias, no lugar de 0 ou de 1, permite o uso de *bleaching*, que ameniza a decisão aleatória em caso de empate, como descrito na seção 3.2.2.

Além da variação do modelo de rede e do modelo de mapeamento dos dados, outros parâmetros foram avaliados, tais como: i) padrões de entrada para as distâncias de Hamming com 8 ou 32 *bits*; ii) variação da janela temporal; iii) padrões de entrada para a rede iguais a 8, 12, 16 ou 32 *bits* vezes  $m$  (número de atributos selecionados); iv) número de entradas para as RAM igual a 2, 3, 4, 6 ou 8 *bits* [Figura 3.1(b)]; v) número de RAM por discriminadores igual à divisão exata do item (iii) pelo item (iv); vi) uso do *bleaching* simples ou percentual; vii) dimensionalidade da aplicação; viii) método de seleção de atributos (RecPun ou RecPun modificado); e ix) quantidade de atributos selecionados (1, 2, ...,  $n$ ). Os itens vii, viii e ix referem-se a aplicações multivariáveis, não se aplicando aos casos univariáveis.

### 3.3.4 Filtro de Agrupamento

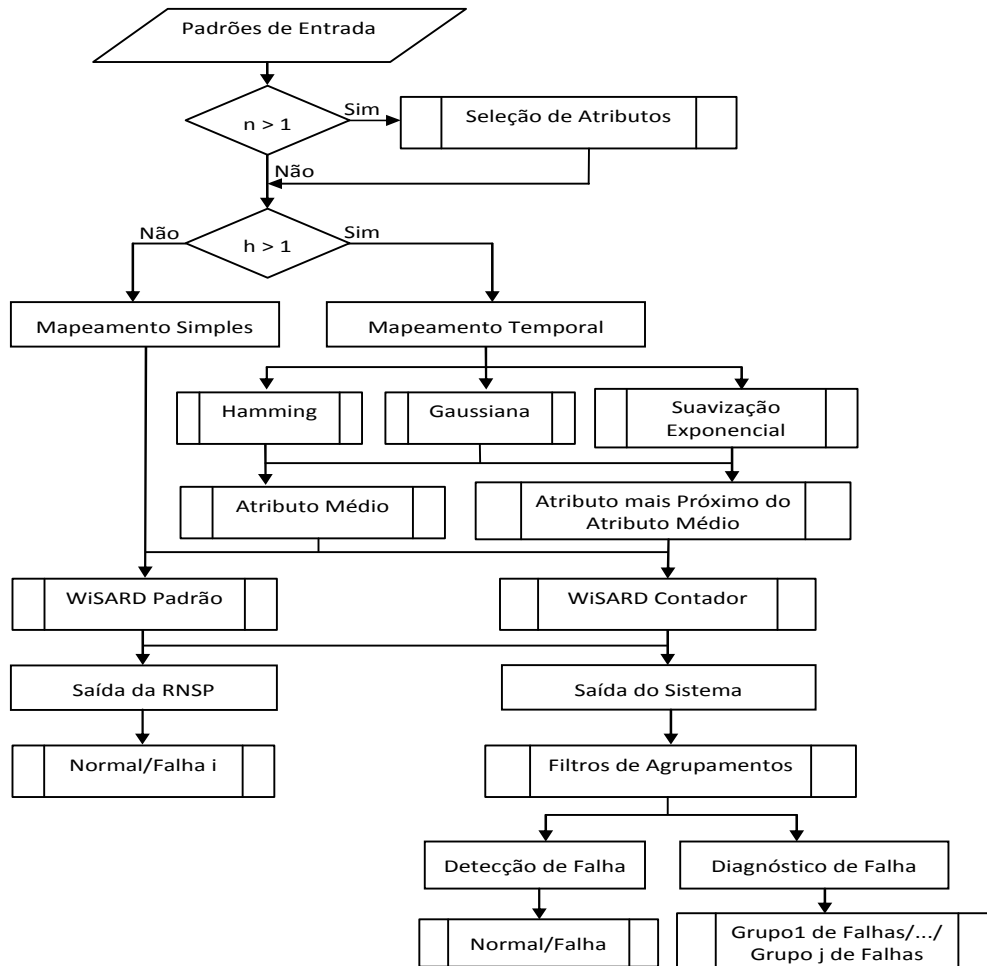
Conforme já mencionado anteriormente, o reconhecimento e a classificação de um padrão de teste são realizados por um dos  $k$  discriminadores que compõem a rede WiSARD. A rede WiSARD equivale a uma estrutura com multiclassificadores, capaz de realiza a detecção e o diagnóstico de falhas de forma paralela e em único procedimento, com um diagnóstico por classe, conforme mostrado na Figura 3.5(a). No entanto, opcionalmente e a partir dos resultados obtidos pela RNSP, se uma classificação por grupo de falhas for requerida, a exemplo dos esquemas ilustrados nas Figuras 3.5(b) e 3.5(c), o filtro de agrupamento como etapa final do sistema FDD-WiSARD é usado. O filtro agrupa apenas as falhas discriminadas em seus respectivos grupos. Aqui dois filtros são aplicados: um para classificação em estado normal ou falha (desvio e rompido); e outro para classificação em estado normal ou grupos de falhas [neste caso, grupos de desvio (D1 e D2) ou rompidos (B1 a B6)]. Dessa forma, é possível uma análise dos resultados em nível da rede (diagnóstico mais granulado) ou do sistema como um todo, a exemplo dos resultados e análises apresentados nas seções 3.4.1.1 e 3.4.2.1.



**Figura 3.5:** Metodologia para os Treinamento e Testes com o Sistema FDD-WiSARD.

### 3.3.5 Metodologia para o Treinamento e Teste do Sistema FDD-WiSARD

A Figura 3.6 sintetiza a metodologia utilizada para a detecção e o diagnóstico de falhas com o sistema FDD-WiSARD, onde  $n$  é o número de atributos e  $h$  é o horizonte da janela temporal. Conforme já mencionado anteriormente, a etapa de seleção de atributos no pré-processamento dos dados só é executada para aplicações multivariáveis. No mapeamento temporal, foram testados os seis modelos (“1” a “6”) de pré-processamento das séries temporais formadas para cada atributo que compõe o padrão de entrada  $x_t$ , conforme descritos na seção 3.3.2.2.



**Figura 3.6:** Metodologia para os Treinamento e Testes com o Sistema FDD-WiSARD.

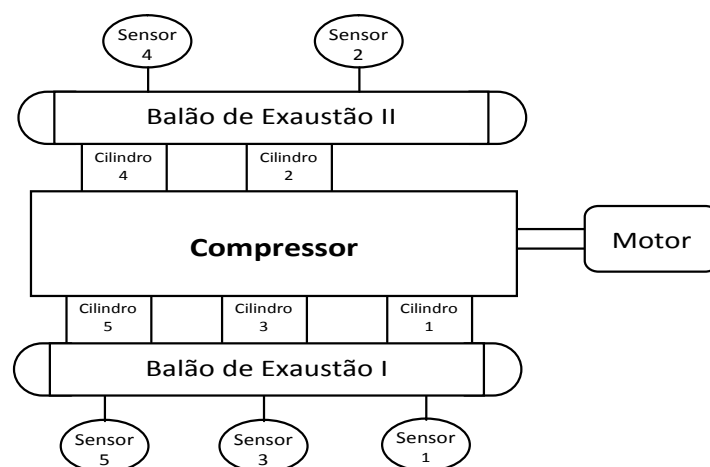
Uma vez que o mapeamento é realizado, o padrão comportamental resultante é apresentado à RNSP. Para avaliar a importância do mecanismo do “bleaching” na rede WiSARD foram testados os modelos WP e WC, conforme descritos na seção 3.3.3. Assim, os resultados dos conjuntos de dados foram avaliados em nível da rede ou do sistema. Como a rede possui um discriminador por classe, consequentemente a saída corresponde a um diagnóstico com os percentuais de acertos para o estado de operação normal e para cada tipo de falha considerada. Porém, outros agrupamentos poderão ser obtidos com a aplicação de um filtro de agrupamento específico na etapa final do sistema FDD-WiSARD. Como a detecção de falhas é realizada pela RNSP e simultaneamente ao diagnóstico, um agrupamento de todas as falhas em um único grupo, classifica os padrões como estado de operação Normal ou estado de Falha, aqui referenciado como detecção de falhas, que na maioria dos sistemas de FDD é realizada em etapa anterior ao diagnóstico.

Na análise de desempenho do sistema, os resultados são apresentados em termos de quantidade e taxa de acerto por classe de padrões. A representação dos resultados apresentados pelos sistemas FDD desenvolvidos é feita por uma matriz de classificação com conteúdos variados. A escolha do sistema final para cada estudo de caso utilizou como critério a exatidão do sistema na detecção e no diagnóstico, para cada aplicação. Os percentuais de acertos/erros referem-se às classificações devidas/indevidas dos dados do conjunto de teste, com base nos limites estabelecidos para as classes consideradas. A acurácia dos sistemas, tanto para a detecção de falhas quanto para o diagnóstico, é obtida na saída do sistema como um todo (rede acoplada ao filtro de agrupamento), conforme mostrado nas Tabelas 3.4, 3.5, 3.12 e 3.13.

### 3.4 Estudos de Caso

#### 3.4.1 Unidade de Processamento de Gás Natural (UPGN)

Para este estudo de caso foi utilizada uma base de dados univariáveis com medições de temperaturas obtidas em uma planta industrial. Os sensores de temperatura estão acoplados a um compressor de gás para a venda (SGC, *Sale Gas Compressor*) de uma Unidade de Processamento de Gás Natural (UPGN) da Petrobras, localizada no município de Pojuca-BA, Brasil (Andrade *et al.*, 2011; Andrade, 2012; Sartori, 2012). Esse compressor trabalha com uma carga mínima de 50% e possui na sua estrutura funcional um motor elétrico, um cárter, cinco cilindros de compressão, dois balões de admissão, dois balões de exaustão e cinco sensores de medição de temperatura, um para cada cilindro, conforme esquema apresentado na Figura 3.7. No referido esquema estão representados somente os balões de exaustão, pois os de admissão são sobrepostos a estes. Os sensores de temperatura monitoram a segurança do sistema e a detecção de possíveis problemas mecânicos nos cilindros de compressão. As medições de temperatura pelos sensores estão numa escala de leitura de 0 °C a 200 °C e são consideradas normais quando assumem valores próximos entre si e no intervalo entre 80 °C e 100 °C, e são consideradas altas e muito altas quando ultrapassam os limites de 140 °C e 160 °C, respectivamente. O sistema de monitoramento é programado para acionar o alarme TAH (*Temperature Alarm High*) ao atingir o limite de 140 °C e o alarme TAHH (*Temperature Alarm High High*) no limite de 160 °C, sendo que o acionamento do segundo alarme provoca a parada do compressor. Além das funções de condicionamento e compressão do gás industrial para o transporte às companhias locais, o SGC fornece energia cinética ao gás para restabelecer sua pressão operacional após a perda que ocorre nas etapas de resfriamento e fracionamento.



**Figura 3.7:** Diagrama Estrutural do Compressor de Gás para a Venda na UPGN em Estudo [adaptado de Andrade (2012)].



Em colaboração com os engenheiros e operadores da planta, as seguintes etapas foram tomadas para construir o banco de dados: “coleta de dados”; “identificação dos padrões de falha”; e “extração das amostras dos padrões identificados” (Andrade *et al.*, 2011; Andrade, 2012; Sartori, 2012). Na “coleta de dados”, as medidas de temperatura foram registradas por um sistema de gerenciamento de informação da planta (PIMS, *Plant Information Management System*). Os intervalos de amostragem foram de 30 segundos, tempo mínimo de armazenamento do PIMS. Ao todo, foram coletadas aproximadamente 600.000 medições de temperaturas, considerando os cinco sensores. Essas medidas correspondem a 208 dias, de modo que quase 7 meses da operação da planta foram monitorados, o que evidencia a robustez do método aqui apresentado. A “identificação dos padrões de falha” consistiu na seleção e caracterização das falhas possíveis. Essas falhas representam o comportamento apresentado pelos sensores nos seus diferentes estados. Os padrões foram selecionados com base no histórico de operação para os estados estacionário e transiente, não sendo considerados os casos de partida e parada de planta. Visando atender as necessidades de desenvolvimento e treinamento das redes neurais, na etapa de “extração das amostras dos padrões identificados” foram obtidos dois conjuntos de dados para o treinamento e a validação dos sistemas desenvolvidos.

Os dados dos conjuntos de treinamento e validação foram rotulados como: “Padrão Normal (*N*)”, para representar as medidas obtidas com o funcionamento normal do sensor e consideradas corretas; ou “Desvio 1 (*D1*)” e “Desvio 2 (*D2*)”, para o estado de medições com desvio; ou “Rompido 1 a 6 (*R1, R2, R3, R4, R5, R6*)”, para a situação em que o sensor encontra-se rompido. As falhas do tipo “Desvio” acontecem quando os estados normais são deslocados para valores maiores do que 100 °C e menores do que 130 °C. Embora sejam consideradas como falhas, não há acionamento de alarmes nem a parada da planta. Já as falhas consideradas como “Rompido” são de alto risco e acontecem por conta de níveis elevados de vibrações no compressor. Essas vibrações em níveis elevados podem ocasionar o rompimento de uma ou mais hastes de transmissão de informações para os sensores e, conseqüentemente, a parada do compressor, após acionamento do alarme TAHH. Embora outros motivos possam provocar os mesmos estados de falha, a vibração do compressor é considerada como a causa principal para a ocorrência dessas falhas. As condições para a ocorrência das falhas são os desgastes do elemento de medição e da haste do sensor, tendo como consequência eventos/defeitos que se iniciam com o afastamento da condição normal e provocam um *offset* (desvio sistemático), podendo chegar ao rompimento da haste do sensor (estado Rompido). As falhas do tipo Rompido são especificadas como: *R1*, mudanças abruptas nas leituras do sensor rompido para o topo de escala (200 °C), com disparo do TAHH (Figuras 3.8(g e h)); *R2*, leituras com valores de fundo (0 °) ou de topo (200 °C) da escala (Figuras 3.8(i e j)); *R3*, mudanças abruptas nas leituras para o fundo de escala (0 °C) (Figuras 3.8(k e l)); *R4*, variação em torno do valor normal de operação (Figuras 3.8(m e n)), sem o rompimento completo das hastes e com transmissão de dados de leitura para o sistema supervisor; *R5*, variações com valores próximos a 0 °C, mas sem atingir o fundo de escala (Figuras 3.8(o e p)); *R6*, variação com valores próximos a 200 °C, mas sem atingir o topo de escala (Figura 3.8(q e r)). Para evitar esses estados indesejados, o sistema desenvolvido procura diagnosticar ao longo do tempo o funcionamento Normal do sensor, a situação do sensor com Desvio sistemático na leitura e a situação de sensor Rompido.

### 3.4.1.1 Resultados e Análises - UPGN

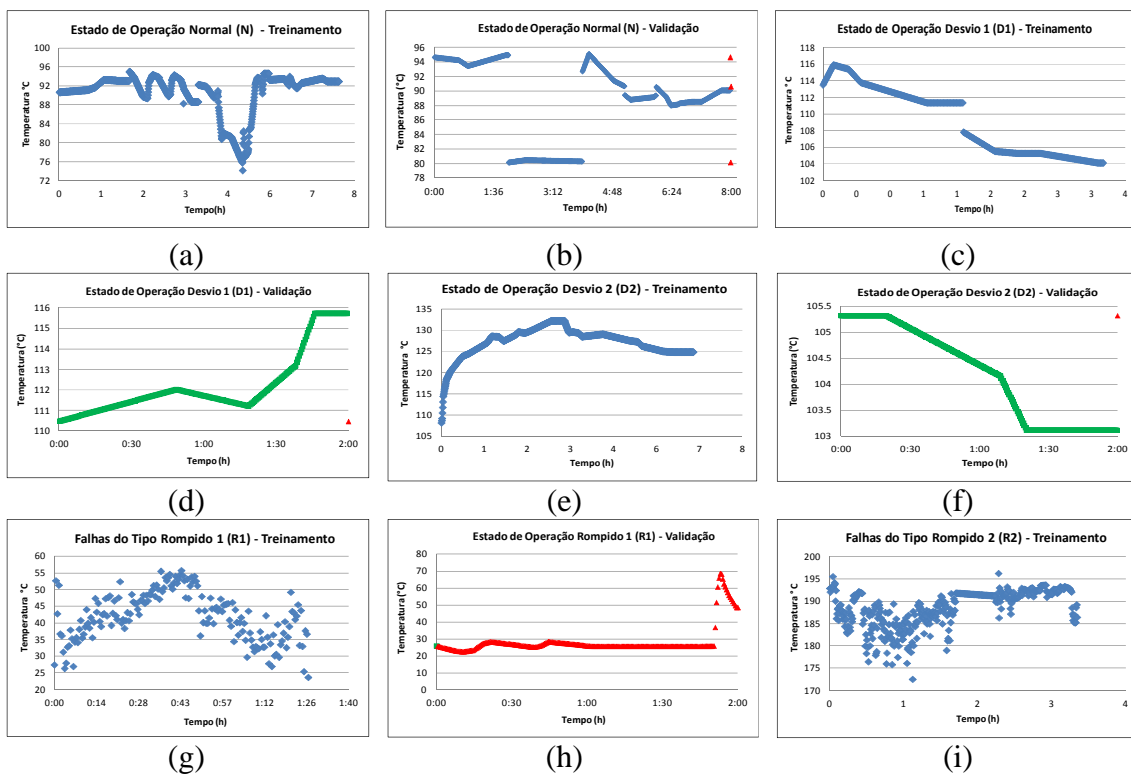
Conforme mencionado anteriormente, o sistema aqui proposto realiza a detecção de falhas como um problema de classificação no qual um padrão de entrada é considerado como estado de “operação normal” ou de “falha”. De modo complementar, o diagnóstico consiste na partição do estado de falha em subclasses com rótulos definidos pelos diferentes tipos de falha considerados. Por se tratar de um estudo de caso univariável, a etapa de seleção de atributos

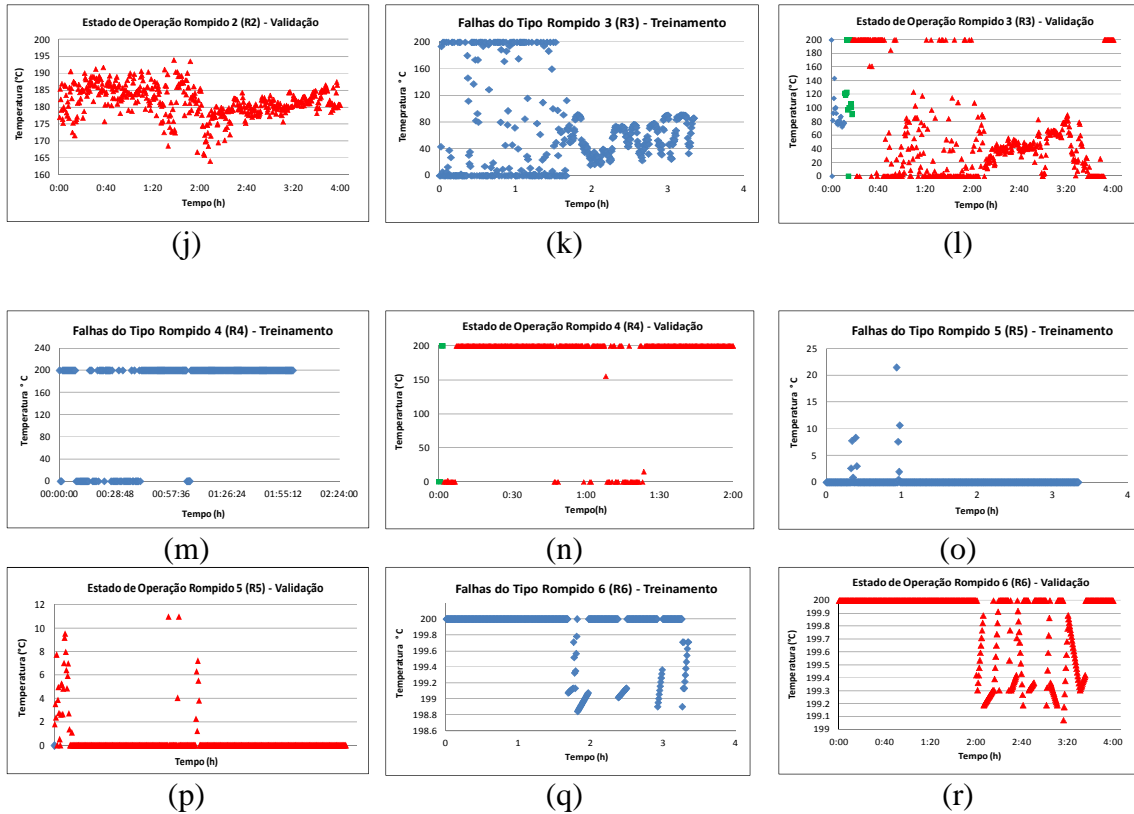
não é executada, como já mencionado na seção 3.3.1. Assim, o pré-processamento dos dados resulta na execução do modelo de “mapeamento simples” ou em um dos seis modelos de “mapeamento temporal”, descritos na seção 3.3.2.2, considerando os demais parâmetros citados na seção 3.3.3. Considerando o período de amostragem de 30 segundos, a Tabela 3.1 mostra o número de pontos de temperatura e o tempo total de operação de cada padrão.

**Tabela 3.1:** Amostras dos Dados de Treinamento e Validação do Sistema FDD-WiSARD

Rótulos/(Classes)	Treinamento			Validação		
	Tempo(h)	Nº Pontos	Figura	Tempo(h)	Nº Pontos	Figura
Normal (N)	8,0	954	7(a)	8,0	961	7(b)
Desvio 1(D1)	4,0	482	7(c)	2,0	241	7(d)
Desvio 2 (D2)	7,0	844	7(e)	2,0	241	7(f)
Rompido 1(R1)	1,5	175	7(g)	2,0	241	7(h)
Rompido 2(R2)	4,0	482	7(i)	4,0	482	7(j)
Rompido 3(R3)	4,0	482	7(k)	4,0	483	7(l)
Rompido 4(R4)	2,0	241	7(m)	2,0	241	7(n)
Rompido 5(R5)	4,0	482	7(o)	2,0	478	7(p)
Rompido 6(R6)	4,0	482	7(q)	2,0	482	7(r)
<b>Total</b>	<b>38,5</b>	<b>4624</b>		<b>28,0</b>	<b>3850</b>	

A título de ilustração, a Figura 3.8 mostra o conjunto de dados originais usados no treinamento e validação dos sistemas para os diferentes modos de operação. Os resultados da validação correspondem à configuração de melhor desempenho do sistema FDD-WiSARD. A Figura 3.8(a) e 3.8(b) referem-se aos dados de treinamento e validação para o modo de operação normal. As demais Figuras referem-se aos estados Desvio (D1 e D2) e falhas do tipo Rompido (R1 a R6).





**Figura 3.8:** Dados Utilizados no Treinamento e Validação dos Sistemas FDD para o caso UPGN. Na Validação: Entradas com Saídas Classificadas com Normal; Entradas com Saídas Classificadas com Desvio; Entradas com Saídas Classificadas com Rompido.

A Tabela 3.2 resume os percentuais de acertos para a detecção e o diagnóstico de falhas, obtidos pelas redes WP e WC, para cada modelo de mapeamento dos padrões de entrada. Considerando ainda os parâmetros discutidos na seção 3.3.3, a Tabela 3.2 mostra os resultados de melhor desempenho obtido na validação dos sistemas para cada par RNSP/modelo de mapeamento. Com base na diversidade de experimentos realizados, a Tabela 3.3 apresenta o conjunto de parâmetros usados com a configuração de melhor desempenho obtido com o sistema FDD-WiSARD para cada par RNSP/modelo de mapeamento mostrado da Tabela 3.2.

**Tabela 3.2:** Resultados da Detecção e do Diagnóstico de Falhas pelo Sistema FDD-WiSARD com Base nos Modelos de Pré-Processamento para as Redes WP e WC

Modelo de Pré-processamento*	Rede WiSARD Padrão (WP)				Rede WiSARD Contador (WC)			
	Detecção de Falhas (%)		Diagnóstico de Falhas (%)		Detecção de Falhas (%)		Diagnóstico de Falhas (%)	
	Normal	Falha	Desvio	Rompido	Normal	Falha	Desvio	Rompido
0	50,05	99,45	58,38	98,84	100,00	98,96	100,00	97,93
1	21,23	94,05	31,33	86,79	98,44	99,13	91,08	97,42
2	49,53	99,30	45,02	97,56	<b>99,69</b>	<b>99,17</b>	<b>99,59</b>	<b>98,26</b>
3	22,79	85,19	31,53	56,75	93,34	89,49	65,35	85,64
4	49,22	99,20	60,58	98,67	99,37	99,10	99,17	98,17
5	21,12	91,19	30,29	83,39	98,96	98,44	60,17	96,47
6	50,26	99,20	56,02	98,05	99,69	98,93	99,17	97,34

\*0) Mapeamento simples; 1) Hamming/padrão médio; 2) Hamming/padrão próximo da média; 3) Gaussiana/padrão médio; 4) Gaussiana/padrão próximo da média; 5) SES/padrão médio; 6) SES/padrão próximo da média.

Como observado na Tabela 3.2, a rede WP apresentou um desempenho levemente superior na detecção de padrões do estado falha em 4 (quatro) dos 7 (sete) modelos de mapeamento utilizados. No entanto, os percentuais de detecção para os padrões do estado de operação Normal e o diagnóstico de falhas foram significativamente inferiores. Como a exatidão do sistema desenvolvido perpassa pela detecção e diagnóstico de falhas, a rede WC com o modelo de mapeamento 2 (dois) (WC-2) foi a que apresentou o melhor desempenho para o sistema FDD-WiSARD. A Tabela 3.4 detalha os resultados obtidos por esta rede.

**Tabela 3.3:** Parâmetros de Melhor Desempenho do Sistema FDD-WiSARD para cada par RNSP/Modelo de Mapeamento Mostrado da Tabela 3.2

Modelo Mapeamento	Rede WiSARD Padrão (WP) *						Rede WiSARD Contador (WC) *					
	i	ii	iii	iv	v	vi	i	ii	iii	iv	v	vi
0	-	1	12	3	4	-	-	1	12	3	4	1
1	8	5	12	3	4	-	32	5	12	3	4	1
2	32	35	12	3	4	-	8	5	12	3	4	1
3	-	5	12	3	4	-	-	10	12	3	4	1
4	-	5	12	3	4	-	-	5	12	3	4	1
5	-	10	12	3	4	-	-	5	12	3	4	1
6	-	40	12	3	4	-	-	5	12	3	4	1

\*i) tamanho-padrão/mapeamentos-Hamming (n° bits); ii) janela-temporal; iii) padrão-entrada-rede (n° bits); iv) entradas-RAM (n° bits); v) n° RAM por discriminador; vi) *bleaching*-simples(1)/ percentual(2).

**Tabela 3.4:** Detecção e Diagnóstico de Falhas do Sistema FDD-WiSARD com o Modelo de Mapeamento 2 e Rede WiSARD Contador (WC-2)

Rótulos/ (Classes)	Detecção de Falhas		Padrões Treinados	Diagnóstico de Falhas	
	Normal	Falha		Desvio	Rompido
N/(1)	958	3	961	0	3
D1/(2)	0	241	241	240	1
D2/(3)	0	241	241	240	1
R1/(4)	0	241	241	1	240
R2/(5)	0	482	482	0	482
R3/(6)	23	460	483	13	447
R4/(7)	0	241	241	4	237
R5/(8)	1	477	478	0	477
R6/(9)	0	482	482	0	482

Conforme já mencionado, as medições de temperatura pelos sensores estão numa escala de leitura de 0 °C a 200 °C e são consideradas normais quando assumem valores próximos entre si e no intervalo entre 80 °C e 100 °C, e são consideradas altas e muito altas quando ultrapassam os respectivos limites de 140 °C (alarme TAH) e 160 °C [alarme TAHH e parada (*trip*) da planta]. Dos padrões rotulados como estado de operação Normal, somente 3 (três) (0.31%) foram detectados e diagnosticados indevidamente como falhas do tipo Rompido, 2 (dois) do tipo R3 e 1 (um) do tipo R6. Neste caso, as classificações indevidas com os padrões Normais não acionariam o sistema de *trip*, por considerar que as informações são falsas, e se a classificação fosse correta o sistema de *trip* também não seria acionado, pois em todos os três casos a temperatura estava em valores abaixo do limite de intertravamento, de modo que essas três classificações incorretas não trariam nenhum problema à operação do sistema. Dos padrões rotulados como estado Desvio (D1 e D2) todos foram detectados corretamente como estado de

falha, embora 1 (um) padrão de *D1* e 1 (um) de *D2* tenham sido diagnosticados de forma indevida como falhas do tipo Rompido, mais especificamente do tipo *R6*. As classificações indevidas com o estado Desvio são menos preocupantes do que aquelas relacionadas ao padrão Normal, pois tanto as informações das classificações corretas Desvio quanto as informações das classificações indevidas apontam para valores de temperaturas não considerados na lógica do trip, a menos que as temperaturas estejam realmente altas e o trip seja acionado de forma preventiva.

Uma vez que não existem classificações indevidas do estado normal para os estados de desvio ou falha do tipo rompido com altas temperaturas, as classificações indevidas com padrões Rompido (*B1*, *B2*, ..., *B6*) são as mais indesejadas, pois valores de temperatura acima de 160 °C indicados por um sensor rompido provocariam a parada indevida da planta. Neste caso, a principal preocupação com o *trip* indevido é econômica, enquanto que o não acionamento do *trip* para os estados Normal e Desvio com temperaturas reais, o que não acontece, causaria acidentes com possíveis danos além de econômicos. Conforme mostrado na Tabela 3.4, dos padrões Rompidos, 24 (vinte e quatro) foram classificados na detecção de falhas como estado de operação Normal, sendo 23 (vinte e três) falhas do tipo *R3* e 1 (uma) do tipo *R5*. Porém, essas classificações incorretas não provocariam a parada indevida da planta, pois os valores de temperatura são inferiores a 160 °C. Já no diagnóstico de falhas, 18 (dezoito) padrões Rompidos foram classificados incorretamente como estado Desvio. Desses, 3 (três) padrões de falhas do tipo *R3* (de um total de 13) e 2 (dois) do tipo *R4* (de um total de 4) apresentaram temperaturas acima de 160 °C, o que provocaria a parada indevida da planta. Porém, como esses 5 (cinco) padrões foram classificados como estado Desvio, o correto seria descartar essas informações do sistema de *trip*, evitando paradas indevidas da planta. Os erros ocorridos com os padrões rompidos *R3* são os mais justificados, visto que as hastes dos sensores não se romperam completamente, havendo possibilidade de transmissão de dados. Assim, os 23 (vinte e três) padrões classificados como estado Normal e os 10 (dez) (de 13) tipo *R3* classificados como Desvio podem ter sido leituras reais (verdadeiras). Embora também indesejados, os erros de classificação dos padrões Rompidos com temperaturas abaixo de 160 °C causam impactos menores, mas importantes na tomada de decisões.

A Tabela 3.5 apresenta uma matriz de confusão típica (Simon e Simon, 2010), também conhecida como matriz de probabilidade de diagnóstico ou matriz de erros (Stehman, 1997), para o conjunto de dados de validação usados nos sistema FDD-WiSARD. Esta representação indica as probabilidades dos estados preditos (possíveis saídas do sistema) para cada classe real (verdadeira).

**Tabela 3.5:** Matriz de Probabilidade de Acerto (Confusão) do Diagnóstico do Sistema Obtido com a Rede WiSARD Contador para o Modelo de Mapeamento 2 (WC-2)

Classificação		Predita		
		Normal (0)	Desvio (1)	Rompido (2)
Real	Normal (0)	<b>0,9969</b>	0,0000	0,0031
	Desvio (1)	0,0000	<b>0,9959</b>	0,0041
	Rompido (2)	0,0100	0,0075	<b>0,9826</b>

As probabilidades de acerto para os três estados são maiores do que 98%. O estado Normal apresentou a maior probabilidade de acerto, sem nenhuma confusão com o estado Desvio e com apenas 3 (três) padrões confundidos com o estado Rompido. O estado Desvio apresentou uma probabilidade de acerto 0.1% menor do que aquela do estado de operação Normal e uma confusão de 2 (dois) padrões com o estado Rompido. Já o estado Rompido foi o que apresentou a menor probabilidade de acerto (98,26%) e uma confusão de 24 (vinte e quatro) padrões com

o estado Normal e 18 (dezoito) com o estado Desvio. Com base nos percentuais de acerto para os estados Normal, Desvio e Rompido, a exatidão total do sistema FDD-WiSARD é de 98,78%.

Considerando apenas a detecção de falhas, isto é, estados de falha (Condição Positiva, CP), compostos pelo desvio e estados rompidos, e estados sem falhas (Condição Negativa, CN), compostos do estado normal, a Tabela 3.6 mostra a matriz de confusão 2x2 para o conjunto de dados de validação utilizado no sistema FDD-WiSARD acompanhado de vários indicadores derivados dele, o que fornece uma análise mais confiável do desempenho do sistema desenvolvido (Fawcett, 2006; Powers, 2011; Ting, 2011):

$$PT = CP + CN \quad (3.10)$$

$$P = 100 \cdot \frac{CP}{T} \quad (3.11)$$

$$TVP = 100 \cdot \frac{VP}{CP} \quad (3.12)$$

$$TVN = 100 \cdot \frac{VN}{CN} \quad (3.13)$$

$$VPP = 100 \cdot \frac{VP}{PP} \quad (3.14)$$

$$VNP = 100 \cdot \frac{VN}{PN} \quad (3.15)$$

$$TFN = 100 \cdot \frac{FN}{CP} \quad (3.16)$$

$$TFP = 100 \cdot \frac{FP}{CN} \quad (3.17)$$

$$TFD = 100 \cdot \frac{FP}{PP} \quad (3.18)$$

$$TFO = 100 \cdot \frac{FN}{PN} \quad (3.19)$$

$$A = 100 \cdot \frac{VP + VN}{T} \quad (3.20)$$

$$LR+ = \frac{TVP}{TFP} \quad (3.21)$$

$$LR- = \frac{TFN}{TVN} \quad (3.22)$$

$$TDO = \frac{LR+}{LR-} \quad (3.23)$$

$$F_1 = 2 \cdot \frac{VPP \cdot TVP}{VPP + TVP} \quad (3.24)$$

$$MCC = 100 \cdot \frac{VP \cdot VN - FP \cdot FN}{\sqrt{(VP + FP) \cdot (VP + FN) \cdot (VN + FP) \cdot (VN + FN)}} \quad (3.25)$$

$$BM = TVP + TVN - 100 \quad (3.26)$$

$$MK = VPP + VNP - 100 \quad (3.27)$$

onde PT é população total, CP é condição positiva (número de casos positivos reais nos dados), CN é condição negativa (número de casos negativos reais nos dados), P é prevalência, VP é verdadeiro positivo (sucesso) VN é verdadeiro negativo (rejeição correta), FP é falso positivo (falsos alarmes ou erro tipo I), FN é falso negativo (falha ou erro Tipo II), TVP é taxa verdadeiro positivo (sensibilidade, *recall*, taxa de sucesso ou probabilidade de detecção), TVN é taxa verdadeiro negativo (especificidade, SPC), VPP é valor positivo preditivo (precisão), VNP é valor negativo preditivo, TFN é taxa falso negativo (taxa de falha), TFP é taxa de falso positivo (queda ou Probabilidade de Falso Alarme), TFD é uma taxa de falsa descoberta, TFO é uma taxa de falsa omissão, ACC é precisão/acurácia, LR+ é razão de verossimilhança positiva, LR- é razão de verossimilhança negativa, TDO é taxa de diagnóstico de probabilidades, F<sub>1</sub> é F1 (a média harmônica de precisão e sensibilidade), MCC é o coeficiente de correlação de Matthews, BM é a informação do bookmaker (ou simplesmente a informação), e MK é marcação. Todos os indicadores demonstram o desempenho bem sucedido do sistema desenvolvido, que tem altas taxas de sucesso e baixas taxas de erro em todas as classes.

**Tabela 3.6:** Matriz de Confusão da Detecção do Sistema Obtida com a Rede WiSARD Contador para o Mapeamento do Modelo 2 (CW-2)

		Condição Predit		Prevalência $P = 75,0\%$	
		Predição Positiva $PP = 2868$	Predição Negativa $PN = 982$		
Condição Verdadeira	População Total $T = 3850$				
	Condição Positiva $CP = 2889$	Verdadeiro Positivo $VP = 2865$	Falso Negativo (erro tipo II) $FN = 24$	Taxa de Verdadeiro Positivo $TVP = 99,2\%$	Taxa de Falso Negativo $TFN = 0,8\%$
	Condição Negativa $CN = 961$	Falso Positivo (erro tipo I) $FP = 3$	Verdadeiro Negativo $VN = 958$	Taxa de Falso Positivo $TFP = 0,3\%$	Taxa de Verdadeiro Negativo $TVN = 99,7$
	ACCurácia $ACC = 99,3\%$ Pontuação F1 $F_1 = 99,5\%$ Bookmaker de Informação $BM = 98,9\%$	Valor Positivo Preditivo $VPP = 99,9\%$	Taxa de Falsa Omissão $TFO = 2,4\%$	Razão de Verossimilhança Positiva $LR+ = 317,7$	Taxa de Diagnostico de Probabilidades $TDO = 38120,4$ Coeficiente de Correlação de Matthews $MCC = 98,2\%$ Marcação $MK = 97,5\%$
	Taxa de Falsa Descoberta $TFD = 0,1\%$	Valor Negativo Preditivo $VNP = 97,6\%$	Razão de Verossimilhança Negativa $LR- = 0,0$		

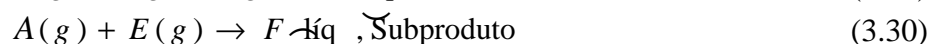
Em comparação, o trabalho desenvolvido por Andrade *et al.*, (2011) também propôs sistemas de detecção e diagnóstico de falhas para esta aplicação, usando o mesmo banco de dados. Eles usaram um algoritmo *Fuzzy C-Means* (FCM) para extrair as regras usadas no sistema de classificação de padrões com base em lógica *fuzzy* tipo-2 intervalar. Os resultados apresentados pelo sistema FDD-WiSARD (Tabela 3.5) e aqueles obtidos por Andrade *et al.*, (2011), considerando a detecção e o diagnóstico de falha para estados de Operação Normal/Desvio/Rompido, são respectivamente: 0,9969/ 0,9959/ 0,9826; e 0,9730/ 0,9970/ 0,9900. A precisão total do sistema proposto por Andrade *et al.*, (2011) foi de 98,66%. Portanto, os resultados obtidos aqui são comparáveis ou até mesmo melhores do que os obtidos por um método diferente das redes neurais sem pesos.

Portanto, o sistema final FDD-WiSARD para a detecção e o diagnóstico de falhas nos sensores de temperatura do compressor de gás para a venda da UPGN em estudo possui a seguinte configuração: “Etapa inicial” - modelo de mapeamento 2 (dois), padrões de entrada para as distâncias de Hamming com 8 (oito) *bits*, horizonte da janela temporal igual a 5 (cinco); “Etapa intermediária” - rede neural sem pesos WiSARD Contador, padrões de entrada para a rede igual a 12 (doze) *bits*, subpadrões de entradas para as RAM com 3 (três) *bits*, número de RAM por discriminador igual a 4 (quatro) (12/3) e uso do *bleaching* simples; “Etapa final” - filtro de agrupamento para a detecção dos estados Normal e de Falha e filtro de agrupamento para o diagnóstico dos estados Desvio e Rompido.

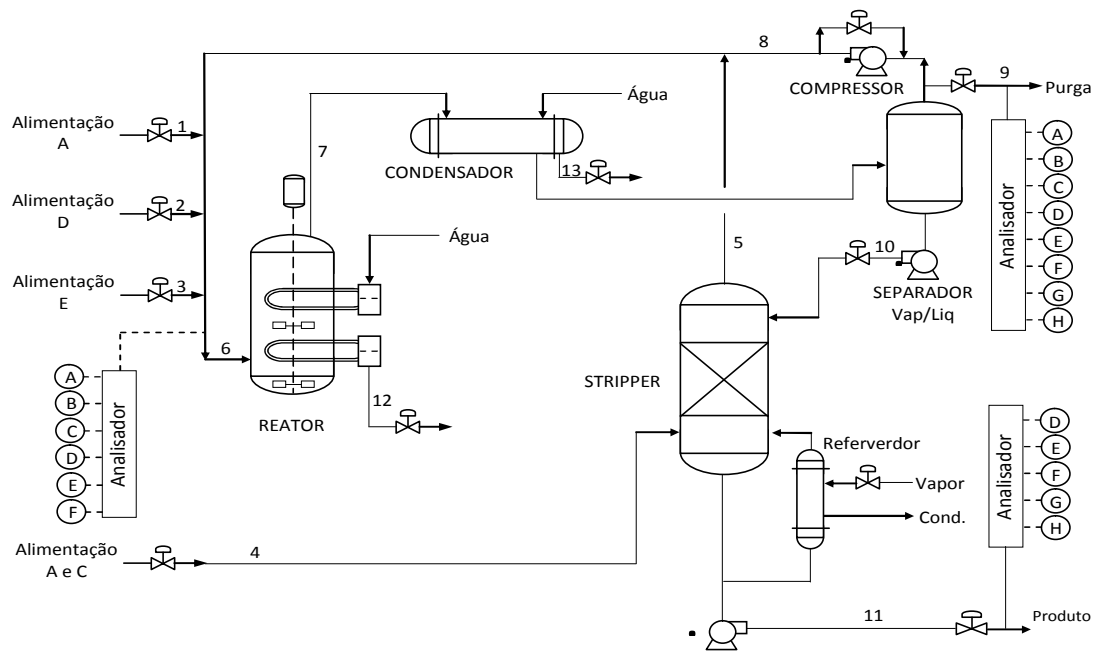
### 3.4.2 O Processo *Tennessee Eastman* (TEP)

O *Tennessee Eastman Process* (TEP) é um modelo de um processo químico tido como referência em estudos sobre controle de processos e, em particular, na área de detecção e diagnóstico de falhas. O TEP foi proposto originalmente por Downs e Vogel (1993) como uma planta industrial em malha aberta com desafios para controle de processos. Posteriormente, surgiram diversos estudos com diferentes técnicas de controle, e aqui é considerada a planta em malha fechada com o sistema de controle proposto por Ricker (1995). Ricker (1995) apresenta ainda seis diferentes estados estacionários para o TEP e um deles foi usado aqui na simulação dos dados para o treinamento do sistema FDD-WiSARD.

Como mostrado na Figura 3.9, o TEP compreende num arranjo reator/separador/reciclo centrado em cinco unidades principais de operação: reator; condensador de produto; separador líquido-vapor; compressor de reciclo; e coluna de separação de produtos. A planta estudada visa a produção dos produtos *G* e *H* a partir dos reagentes *A*, *C*, *D* e *E*, gerando ainda um subproduto *F*, de acordo com as reações representadas pelas Equações 3.28 a 3.30. Além dos reagentes e produtos é utilizado também um inerte gasoso *B*, introduzido nas correntes de alimentação dos reagentes.







**Figura 3.9:** Diagrama Esquemático do *Tennessee Eastman Process* (TEP).

Os reagentes *A*, *D* e *E* são adicionados ao reator e o reagente *C* alimenta a coluna de separação (*stripper*, retificadora). Uma quantidade pequena de um inerte *B*, responsável principal pela purga no sistema, é inserido na alimentação do reagente *C*. A saída do reator é conduzida para o condensador onde ocorre uma liquefação parcial por resfriamento. A saída do condensador segue para a unidade de separação líquido/vapor onde uma fração da parte gasosa é purgada para evitar acumulação de inerte e subproduto no processo e a outra fração é enviada para o compressor centrífugo e reciclada pelo reator. A parte líquida segue para a *stripper* (coluna retificadora), onde uma mistura dos reagentes *A* e *C*, com a ajuda do refeedor, separa o restante dos componentes leves, que também é reciclado pelo reator. A parte líquida, escoada pelo fluxo 11, contém basicamente *G* e *H* e consiste no produto final do processo. A separação dos produtos *G* e *H* não faz parte do TEP (Soares e Galvão, 2010).

O modelo TEP possui 12 variáveis manipuláveis e 41 variáveis medidas. Conforme mostra na Tabela 3.7, das variáveis medidas, 22 são amostradas a cada 3 minutos e correspondem às medições de vazão, temperatura e pressão. As demais variáveis medidas têm taxas de amostragem maiores, e consistem dos analisadores da alimentação para o reator, da purga e do produto final. Para este trabalho, foram consideradas 13 diferentes condições de falhas em malha fechada (Ricker, 1995) e descritas na Tabela 3.8. No trabalho original, Downs e Vogel (1993) analisaram 21 condições de falhas em malha aberta. Alterações do tipo oscilatório consistem em mudanças aleatórias e persistentes na variável considerada.

**Tabela 3.7:** Variáveis Medidas no TEP

Descrição	Grandeza	t (min)	Unidade
Vazão de alimentação para o reagente <i>A</i>	$y_1$	3	kscmh
Vazão de alimentação para o reagente <i>D</i>	$y_2$	3	kg/h
Vazão de alimentação para o reagente <i>E</i>	$y_3$	3	kg/h
Vazão de alimentação para os reagentes <i>A</i> e <i>C</i>	$y_4$	3	kscmh
Vazão de reciclo do compressor	$y_5$	3	kscmh
Vazão de alimentação do reator	$y_6$	3	kscmh
Pressão do reator	$y_7$	3	kPa
Nível do reator	$y_8$	3	%
Temperatura do reator	$y_9$	3	°C
Vazão de purga	$y_{10}$	3	Kscmh
Temperatura do separador de produtos	$y_{11}$	3	°C
Nível do separador de produtos	$y_{12}$	3	%
Pressão do separador de produtos	$y_{13}$	3	kPa
Vazão de líquido na saída do separador	$y_{14}$	3	m <sup>3</sup> /h
Nível da retificadora	$y_{15}$	3	%
Pressão da retificadora	$y_{16}$	3	kPa
Vazão de líquido na saída da retificadora	$y_{17}$	3	m <sup>3</sup> /h
Temperatura da retificadora	$y_{18}$	3	°C
Vazão de vapor para a retificadora	$y_{19}$	3	kg/h
Trabalho do compressor	$y_{20}$	3	kW
Temperatura da água de resfriamento para o reator	$y_{21}$	3	°C
Temperatura da água de resfriamento para o condensador	$y_{22}$	3	°C
Composição de <i>A</i> na alimentação do reator	$y_{23}$	6	mol%
Composição de <i>B</i> na alimentação do reator	$y_{24}$	6	mol%
Composição de <i>C</i> na alimentação do reator	$y_{25}$	6	mol%
Composição de <i>D</i> na alimentação do reator	$y_{26}$	6	mol%
Composição de <i>E</i> na alimentação do reator	$y_{27}$	6	mol%
Composição de <i>F</i> na alimentação do reator	$y_{28}$	6	mol%
Composição de <i>A</i> no gás de purga	$y_{29}$	6	mol%
Composição de <i>B</i> no gás de purga	$y_{30}$	6	mol%
Composição de <i>C</i> no gás de purga	$y_{31}$	6	mol%
Composição de <i>D</i> no gás de purga	$y_{32}$	6	mol%
Composição de <i>E</i> no gás de purga	$y_{33}$	6	mol%
Composição de <i>F</i> no gás de purga	$y_{34}$	6	mol%
Composição de <i>G</i> no gás de purga	$y_{35}$	6	mol%
Composição de <i>H</i> no gás de purga	$y_{36}$	6	mol%
Composição de <i>D</i> na retificadora	$y_{37}$	15	mol%
Composição de <i>E</i> na retificadora	$y_{38}$	15	mol%
Composição de <i>F</i> na retificadora	$y_{39}$	15	mol%
Composição de <i>G</i> na retificadora	$y_{40}$	15	mol%
Composição de <i>H</i> na retificadora	$y_{41}$	15	mol%

**Tabela 3.8:** Falhas Simuladas no TEP

<b>Falha</b>	<b>Descrição</b>	<b>Alteração</b>
1	Razão <i>A/C</i> , mantendo <i>B</i> constante	Degrau
2	Composição de <i>B</i> , mantendo a razão <i>A/C</i> constante	Degrau
3	Temperatura de alimentação de <i>D</i>	Degrau
4	Temperatura da água de refrigeração para o reator	Degrau
5	Temperatura da água de refrigeração para o condensador	Degrau
6	Carga (válvula) de alimentação de <i>A</i>	Degrau
7	Carga (válvula) de alimentação de <i>C</i>	Oscilatória
8	Composições das alimentações de <i>A</i> , <i>B</i> e <i>C</i>	Oscilatória
9	Temperatura de alimentação de <i>A</i>	Oscilatória
10	Temperatura de alimentação de <i>C</i>	Oscilatória
11	Temperatura da água de refrigeração para o reator	Oscilatória
12	Temperatura da água de refrigeração para o condensador	Oscilatória
13	Variação na cinética da reação (velocidade, componentes e fatores)	Oscilatória

### 3.4.2.1 Resultados e Análises - TEP

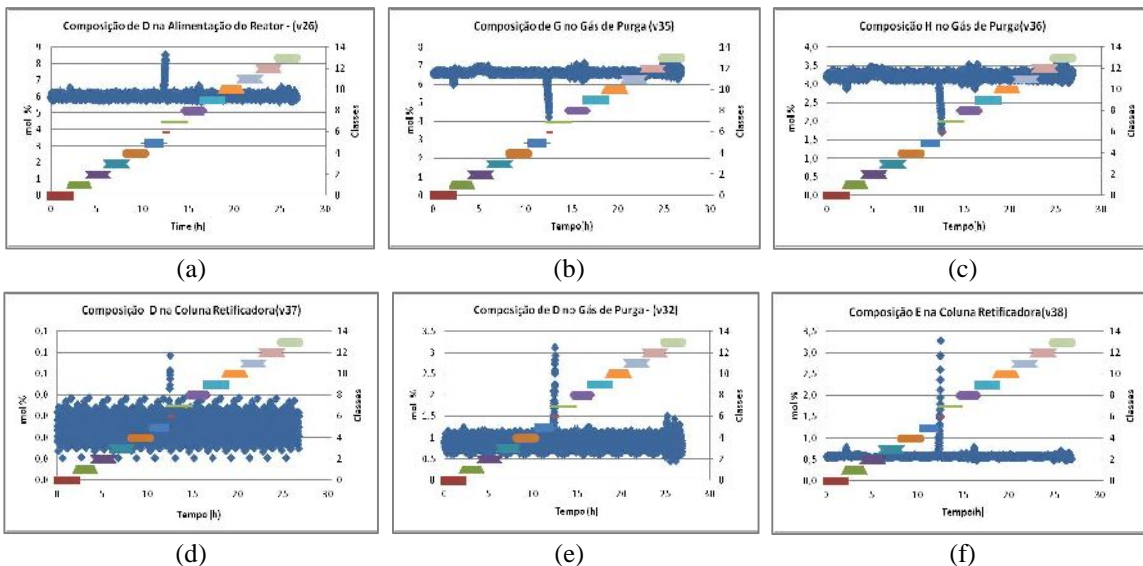
Os dados para treinamento e validação dos sistemas de FDD usados no TEP foram gerados através do simulador proposto por Ricker (1995), com o uso dos valores referenciados para o modo de operação “caso 1” proposto pelo autor. Foram consideradas 5 (cinco) diferentes condições, uma com os valores originais, que foi usada para o treinamento das redes, e outras 4 (quatro) que foram obtidas com variações de  $\pm 3\%$  e  $\pm 5\%$  na variável taxa de produção e usadas na validação dos sistemas. Ricker (1995) considera essa variável controlada como a de maior relevância na composição dos produtos finais *G* e *H* e admite como aceitáveis variações no intervalo  $\pm 5\%$ . Com exceção da falha 6, para cada configuração foram realizadas 14 simulações de 1000 pontos de amostragem de 3 minutos, sendo uma simulação de operação normal e as outras referentes aos 12 tipos de falha considerados. Com o intuito de amenizar os efeitos de inicialização dos sistemas foram descartados os 20 primeiros períodos de cada simulação, conforme detalhes mostrados na Tabela 3.9.

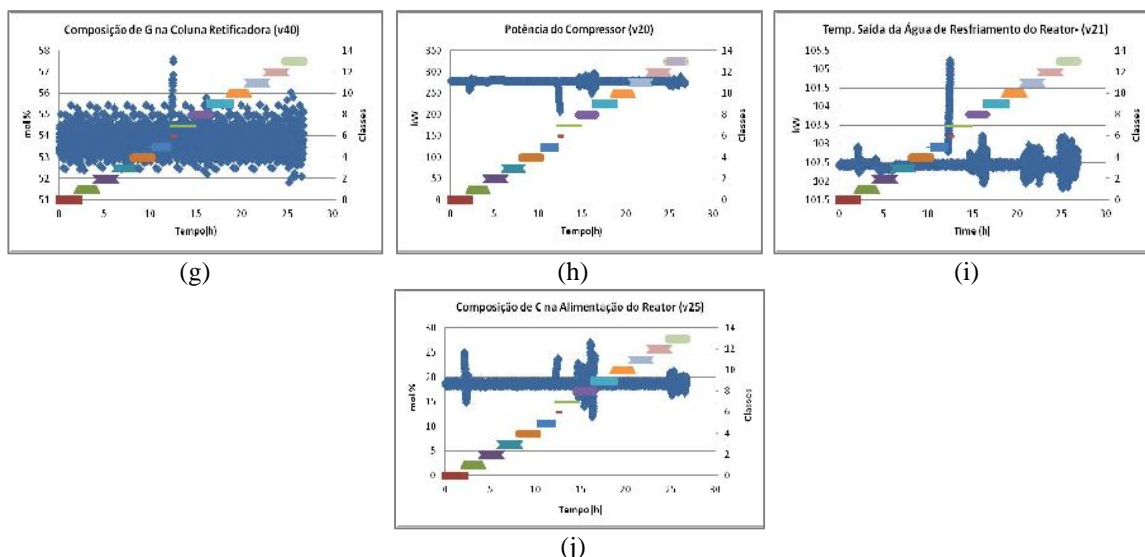
**Tabela 3.9:** Amostras dos Dados de Treinamento e Validação do Sistema FDD-WiSARD

Classes	Treinamento		Validação**	
	Tempo(h)	Nº Pontos*	Tempo(h)	Nº Pontos*
Normal (N)	49	980	49	980
Falha 1	49	980	49	980
Falha 2	49	980	49	980
Falha 3	49	980	49	980
Falha 4	49	980	49	980
Falha 5	49	980	49	980
Falha 6	6	121	6	116
Falha 7	49	980	49	980
Falha 8	49	980	49	980
Falha 9	49	980	49	980
Falha 10	49	980	49	980
Falha 11	49	980	49	980
Falha 12	49	980	49	980
Falha 13	49	980	49	980
<b>Total</b>	<b>643</b>	<b>12861</b>	<b>643</b>	<b>12856</b>

\*Relativos a cada variável de entrada. \*\* Variação de +3% na taxa de produção.

A título de ilustração, a Figura 3.10 mostra o conjunto de dados originais de treinamento relacionados aos modos de operação “Normal” e “Falha 4”, após seleção de variáveis (atributos) com o algoritmo RecPun modificado e relativos à configuração de melhor desempenho com o sistema FDD-WiSARD. As Figuras 3.10(a) a 3.10(d) são variáveis comuns aos modos de operação Normal e Falha 4, as Figuras 3.10(e) a 3.10(g) são as demais variáveis relativas ao modo de operação Normal e as Figuras 3.10(h) a 3.10(j) complementam as variáveis selecionadas para o estado de Falha 4.





**Figura 3.10:** Dados Utilizados no Treinamento dos Sistemas FDD para o CSTR: Modos de Operação “Normal” e “Falha 4”.

Os resultados mostrados na Tabela 3.10 referem-se aos percentuais de acerto no diagnóstico apresentado pelas RNSP, com o uso prévio do algoritmo de seleção de atributos RecPun modificado. O percentual em cada célula na Tabela 3.10 corresponde ao melhor desempenho para uma dada configuração (modelo de mapeamento *versus* rede *versus* variação na variável taxa de produção), com os demais parâmetros descritos na seção 3.3.3. Diante da diversidade de experimentos realizados, a Tabela 3.11 apresenta o conjunto de parâmetros de melhor desempenho do sistema FDD-WiSARD para cada par RNSP/modelo de mapeamento mostrado da Tabela 3.10.

**Tabela 3.10:** Percentuais de Acerto das RNSP para o TEP (%)

Modelo de Pré-processamento*	WiSARD Padrão (WP)					WiSARD Contador (WC)				
	-5	-3	+3	+5	Média	-5	-3	+3	+5	Média
0	97,35	99,18	99,24	98,15	98,48	98,17	99,32	<b>99,47</b>	98,20	98,79
1	79,43	95,56	91,33	79,77	86,52	93,65	98,38	98,47	94,89	96,35
2	79,68	91,03	83,72	80,47	83,73	92,97	97,57	98,41	93,65	95,65
3	54,27	96,20	94,74	57,87	75,77	90,56	99,07	99,08	93,39	95,53
4	79,59	90,30	89,53	80,37	84,95	93,18	97,48	98,24	93,34	95,56
5	84,20	90,14	93,12	81,03	87,12	94,86	98,70	98,92	94,78	96,82
6	87,02	98,51	97,00	88,36	92,72	94,88	98,21	99,12	96,66	97,22

\*0) Mapeamento simples; 1) Hamming/padrão médio; 2) Hamming/padrão próximo da média; 3) Gaussiana/padrão médio; 4) Gaussiana/padrão próximo da média; 5) SES/padrão médio; 6) SES/padrão próximo da média.

**Tabela 3.11:** Parâmetros de Melhor Desempenho do Sistema FDD-WiSARD para cada par RNSP/Modelo de Mapeamento Mostrado da Tabela 3.10

Modelo Mapeamento	Rede WiSARD Padrão (WP)*							Rede WiSARD Contador (WC)*						
	i	ii	iii	iv	v	vi	vii	i	ii	iii	iv	v	vi	vii
0	-	1	84	3	28	-	7	-	1	84	3	28	2	7
1	8	5	84	3	28	-	7	8	5	84	3	28	2	7
2	8	10	84	3	28	-	7	8	5	84	3	28	1	7
3	-	5	96	3	32	-	8	-	5	84	3	28	1	7
4	-	5	96	3	32	-	8	-	5	84	3	28	1	7
5	-	5	84	3	28	-	7	-	5	84	3	28	2	7
6	-	5	96	3	32	-	8	-	5	84	3	28	2	7

i) tamanho padrão/ mapeamentos de Hamming (n. bits); ii) janela temporal; iii) padrão de entrada da rede (n. bits); iv) entradas RAM (n. bits); v) n. de RAM por discriminadores; vi) *bleaching* simples (1)/ percentual (2); e vii) quantidade de atributos selecionados.

Com base nos dados da Tabela 3.10 as seguintes considerações podem ser feitas:

- A rede WC apresentou melhor desempenho com diferenças significativas em relação à WP. O baixo rendimento na rede WP justifica-se pelo fato dela guardar apenas o último aprendizado nos endereços de memória acessados. Além disso, o uso do critério de aleatoriedade para o desempate dos discriminadores com somatórios de valores 1 maior do que 0 (zero) diminui o desempenho da rede;
- O desempenho dos sistemas de FDD é inversamente proporcional ao percentual de variação na variável taxa de produção, o que justifica a recomendação dada por Ricker (1995) em não usar variações fora do intervalo de  $\pm 5\%$ ;
- O modelo de mapeamento 0 (zero) com a rede WC (WC-0) foi a configuração com melhor desempenho de diagnóstico, com uma média de acerto igual a 98,79% e com o melhor desempenho igual 99,47% (variação de +3% na taxa de produção) e o pior igual a 98,17% (variação de -5% na taxa de produção). Embora o mapeamento 0 (zero) não utilize a janela temporal no pré-processamento dos dados, o comportamento dinâmico na rede WC é devido aos incrementos nos valores escalares armazenados nos endereços de memória. Os valores armazenados, nos mesmos endereços ou em endereços próximos, registram comportamentos semelhantes existentes nos dados históricos do problema.

A matriz de classificação apresentada na Tabela 3.12 detalha o diagnóstico da configuração WC-0 (com variação de +3% na taxa de produção) por classe obtido na saída da rede e o uso do algoritmo de seleção de atributos “RecPun modificado” na etapa de pré-processamento dos dados. De modo comparativo, a matriz de classificação apresentada na Tabela 3.13 detalha o mesmo diagnóstico, porém com o uso do algoritmo de seleção de atributos “RecPun original”.

**Tabela 3.12:** Matriz de Classificação (Confusão) para a Rede WiSARD Contador (WC-0) com Variação de +3% na Taxa de Produção: Algoritmo de Seleção de Atributos RecPun Modificado

Rótulos / Classes	Saídas Preditas da RNSP-WC-0														Acertos (%)
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	
C0	980														100
C1		970					10								98,98
C2			980												100
C3				975					5						99,49
C4					957	23									97,65
C5						980									100
C6							115							1	99,14
C7								980							100
C8									980						100
C9		4								975				1	99,49
C10											980				100
C11												980			100
C12		7				4			12				957		97,65
C13							1							979	99,90
<b>Percentual de Acertos da Rede</b>															<b>99,47</b>

**Tabela 3.13:** Matriz de Classificação (Confusão) para a Rede WiSARD Contador (W-0) com Variação de +3% na Taxa de Produção: Algoritmo de Seleção de Atributos RecPun Original

Classes (Real)	Saídas Preditas pela RNSP-WC														Acertos (%)
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	
C0	10			2	37	20		94		87	483	183	64		1,02
C1		970					10								98,98
C2			980												100
C3	14			3	35	22		93		80	480	189	64		0,31
C4	12				46	13		90		75	475	199	70		4,69
C5	8			3	36	23		85		87	491	179	68		2,35
C6							115							1	99,14
C7	4			2	23	27		86		82	491	186	76	3	8,78
C8									980						100
C9	7			4	34	16		93		79	511	172	62	2	8,06
C10	4			2	21	5		95		82	576	140	55		58,78
C11	9				41	24		110		70	457	206	63		21,02
C12	11			2	46	21		95		78	453	198	76		7,76
C13							1							979	99,90
<b>Percentual de Acertos da Rede</b>															<b>39,90</b>

Comparando as matrizes de classificação nas Tabelas 3.11 e 3.12 percebe-se um desempenho superior do algoritmo RecPun modificado em relação ao algoritmo original. A confusão na

classificação com o RecPun original aconteceu porque os atributos selecionados são os mesmos para a maioria das classes, como mostrado na Tabela 3.14(a). Para amenizar o ocorrido, como já mencionado na seção 3.3.2.1, foi adicionado um passo ao final do algoritmo RecPun de modo que a relação biunívoca entre as classes permita no máximo 50% dos atributos comuns, como mostrado na Tabela 3.14(b).

**Tabela 3.14:** Atributos Selecionados por Classe pelos Algoritmos RecPun Original e RecPun Modificado, para WC-0 com Variação de +3% na Taxa de Produção

Classes	RecPun Original	RecPun Modificado
0	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 38, 40, 32
1	4, 2, 5, 3, 1, 6, 7	4, 2, 5, 3, 1, 6, 7
2	38, 40, 31, 34, 27, 36, 23	38, 40, 31, 34, 27, 36, 23
3	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 39, 34, 27
4	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 20, 25, 21
5	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 22, 31, 18
6	2, 1, 3, 7, 9, 8, 11	2, 1, 3, 7, 9, 8, 11
7	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 23, 28, 19
8	5, 0, 4, 37, 35, 38, 6	5, 0, 4, 37, 35, 38, 6
9	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 24, 33, 30
10	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 21, 25, 22
11	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 22, 25, 31
12	37, 36, 35, 26, 38, 40, 32	37, 36, 35, 26, 29, 17, 0
13	16, 17, 15, 21, 14, 18, 34	16, 17, 15, 21, 14, 18, 34

(a) RecPun Original

(b) RecPun Original

Considerando apenas a detecção de falhas, independentemente do diagnóstico, a Tabela 3.15 mostra a matriz de confusão 2x2 para o conjunto de dados de validação usado no sistema FDD-WiSARD acompanhado de seus indicadores. Como esperado, uma vez que a precisão é de 100%, todos os indicadores funcionam perfeitamente no sistema desenvolvido.



**Tabela 3.15:** Matriz de Confusão da Detecção da Rede Contador WiSARD (CW-0) com variação de + 3% na Taxa de Produção: Algoritmo de Seleção de Atributos RecPun Modificado

		Condições Preditas			
População Total $T = 12856$		Predição Positiva $PP = 11876$	Predição Negativa $PN = 980$	Prevalência $P = 92.4\%$	
Condições Verdadeiras	Condição Positiva $CP = 11876$	Verdadeiro Positivo $VP = 11876$	Falso Negativo (erro tipo II) $FN = 0$	Taxa de Verdadeiro Positivo $TVP = 100,0\%$	Taxa de Falso Negativo $TFN = 0,0\%$
	Condição Negativa $CN = 980$	Falso Positivo (erro tipo I) $FP = 0$	Verdadeiro Negativo $VN = 980$	Taxa de Falso Positivo $TFP = 0,0\%$	Taxa de Verdadeiro Negativo $TVN = 100,0$
ACCurácia $ACC = 100,0\%$ Pontuação F1 $F_1 = 100,0\%$ Bookmaker de Informação $BM = 100,0\%$	Valor Positivo Preditivo $VPP = 100,0\%$	Taxa de Falsa Omissão $TFO = 0,0\%$	Razão de Verossimilhança Positiva $LR+ =$	Taxa de Diagnostico de Probabilidades $TDP =$ Coeficiente de Correlação de Matthews $MCC = 100,0\%$ Marcação $MK = 100,0\%$	
	Taxa de Falsa Descoberta $TFD = 0,0\%$	Valor Negativo Preditivo $VNP = 100,0\%$	Razão de Verossimilhança Negativa $LR- = 0,0$		

Soares e Galvão (2010) examinaram a detecção de falhas na TEP usando classificadores para grupos de falhas com características semelhantes. Para a definição desses grupos de falhas, utilizou-se um algoritmo de aprendizado não supervisionado e o melhor desempenho resultou nos 4 grupos mostrados no final da Tabela 3.16. Esses autores compararam classificadores obtidos através de métodos paramétricos (Análise Discriminante Linear, ADL, com Algoritmo de Projeções Sucessivas, APS, e com Algoritmo Genético, AG, e Análise Discriminante Quadrática, ADQ, com APS) e não paramétricos ( $k$ -vizinho mais próximo, kNN e mínimos quadrados parciais, PLS). O resultado de melhor desempenho foi obtido com os classificadores APS-ADQ, conforme mostrado na Tabela 3.16. Além disso, a Tabela 3.16 mostra um resumo dos principais indicadores de qualidade para o sistema aqui desenvolvido com a configuração CW-0 e variação de + 3% no taxa de produção de  $G$  e  $H$ . O melhor desempenho é comparado com os métodos apresentados por Soares e Galvão (2010).

**Tabela 3.16:** Indicadores de Qualidade na Validação do Sistema FDD-WiSARD (WC-0 com Variação de +3% na Taxa de Produção) e Comparação com Soares e Galvão (2010)

Itens Observados	RecPun Original	RecPun Modificado	(Soares e Galvão, 2010)				
			Paramétricos			Não Paramétricos	
			APS-ADQ	APS-ADL	AG-ADL	kNN	PLS
Padrões Testados	12856	12856	-	-	-		
Acertos - RNSP (n.)	5129	12788	-	-	-		
Acertos - RNSP (%)	39,90%	<b>99,47%</b>	-	-	-		
Filtro 1: Detecção de falhas (Média)	99,42%	<b>100,00%</b>	-	-	-		
• Estado de Operação Normal	1,02%	<b>100,00%</b>	-	-	-		
• Estado de Operação com Falha	99,42%	<b>100,00%</b>	-	-	-		
Filtro 2: Diagnóstico por Grupos de Falhas (Soares & Galvão, 2010)	58,02%	<b>99,43%</b>	93,20%	86,0%	93,2%		
• Grupo 1 (falhas: 7, 8, 9 e 10)	78,44%	<b>99,87%</b>	94,1%	85,0%	76,0%	71,7%	79,3%
• Grupo 2 (falhas: 1 e 4)	51,84%	<b>98,32%</b>	97,8%	90,7%	76,3%	72,2%	77,3%
• Grupo 3 (falhas: 2, 3, 5, 6 e 11)	43,21%	<b>99,85%</b>	94,9%	84,8%	76,2%	74,6%	80,8%
• Grupo 4 (falhas: 12 e 13)	53,83%	<b>98,78%</b>	86,0%	82,8%	72,0%	70,6%	75,9%

Os resultados apresentados por Lau *et al.* (2013) também permitem uma comparação dos resultados aqui obtidos para o estudo de caso TEP. Lau *et al.* (2013) propuseram um sistema de diagnóstico de falhas baseado em *Multi-Scale Principal Component Analysis* (MSPCA) e no *Adaptive Network-based Inference System* (ANFIS) e testaram com o TEP. O referido sistema foi referenciado como MSPCA-ANFIS e composto de duas etapas principais. A primeira realiza a extração de características usando MSPCA e a segunda realiza a classificação de padrões com ANFIS. O ANFIS no modelo MSPCA-ANFIS é um sistema multiclassificador com propósitos semelhantes à WiSARD. No MSPCA-ANFIS, cada classificador ANFIS realiza a detecção e o diagnóstico de um único tipo de falha com base num limiar pré-definido, função semelhante à técnica de *bleaching* na rede WiSARD contador. Os resultados obtidos com o modelo MSPCA-ANFIS foram superiores com relação aos modelos comparados e apresentou um percentual de classificação correta de 99,52% para os padrões de operação normal e um percentual médio de reconhecimento de 90,7% para os estados de falhas das 21 falhas analisadas no processo TEP, excluindo as falhas 3, 9 e 15. No modelo aqui proposto esses percentuais foram de 100%, conforme mostrado na Tabela 3.16.

O sistema de reconhecimento de padrões multivariados apresentado por Barragan *et al.* (2016) compreende num agrupamento de padrões com base no método *Fuzzy C-Means* (FCM). Para isso, é usado um Fator de Similaridade PCA Multiscale (SPCA<sub>ms</sub>) que determina as distâncias

dos objetos (séries temporais multivariadas) a serem reconhecidos. Esse fator é obtido a partir da análise de sinais de algumas variáveis do TEP utilizando a transformada wavelet. A base de dados foi obtida a partir do simulador TEP, proposto por Ricker (1995), e apresenta algumas características peculiares, tais como: contém dados relativos ao estado de operação normal e estado de falha 1; utiliza 4, das 41 variáveis do TEP; eventos de falha e operação normal foram coletados com diferentes mudanças nos modos de operação (estados estacionários). Os padrões de testes são agrupados em dois (Normal ou falha) ou três (Normal, falha ou híbridos) *clusters*. As métricas do SPCA<sub>ms</sub> tiveram uma taxa de classificação de sucesso de 95% para os *clusters* 1 (falha) e 2 (normal), e com SPCA tradicional, essas taxas foram de 70% e 80%, respectivamente.

Portanto, os resultados obtidos aqui são comparáveis ou melhores do que aqueles obtidos com métodos não baseados em RNSP. A partir dos resultados e comparações acima, a estrutura final do sistema FDD-WiSARD para o modelo TEP tem a seguinte configuração: “Etapa inicial” - algoritmo de seleção de atributos RecPun modificado, modelo de mapeamento simples 0 (zero); “Etapa intermediária” - RNSP WiSARD Contador (WC), padrões de entrada para a rede com 84 *bits* (12x7), subpadrões de entradas para as RAM com 3 (três) *bits*, número de RAM por discriminador igual a 28 (84/3), uso do *bleaching* percentual; “Etapa final” - filtro de agrupamento para a detecção do estado normal e de falha e filtro de agrupamento para o diagnóstico por grupos de falhas, de acordo com os definidos por Soares e Galvão (2010) e mostrados na Tabela 3.16. O tempo de execução do treinamento e teste do sistema para a referida configuração foi de 3(três) segundos. Os dados de treinamento e validação foram simulados com base no modo de operação “caso 1” proposto por Ricker (1995) com variação de +3% na taxa de produção para o conjunto de validação.

### 3.5 Conclusões Parciais

Este trabalho aborda o problema da detecção e diagnóstico de falhas em sistemas dinâmicos com o uso de redes neurais sem peso. A rede DW foi utilizada como referência para o desenvolvimento dos sistemas propostos e esses sistemas foram testados e validados com dados univariados, representando um problema real (dados históricos de uma UPGN, com foco no monitoramento de temperatura de seu compressor de gás de vendas) e com dados multivariados simulados de um problema *benchmark*, conhecido de TEP. Com os resultados obtidos para os dois estudos de caso verificou-se que as RNSP apresentaram uma boa solução para o problema de detecção e diagnóstico de falhas em sistemas dinâmicos univariados e multivariados. Isso foi possível dada a contribuição de vários elementos importantes: seleção de atributos (variáveis) por classe como contribuição importante para o pré-processamento de dados multivariados; pré-processamento de dados através de modelos de mapeamento de “simples” e de “séries temporais”, com janelas deslizantes com base em distâncias de Hamming, função de distribuição de probabilidade gaussiana e suavização exponencial simples; o uso da abordagem contador no reconhecimento de padrões e na classificação realizada pela rede WiSARD Contador; o uso da técnica de *bleaching* (simples e incremental) para minimizar o uso de aleatoriedade nas saídas das RNSP; e os filtros de agrupamento (*clustering*) usados para obter o diagnóstico por grupos de falhas, uma vez que a performance da RNSP realiza a detecção e o diagnóstico de falhas por classe (falhas individuais) simultaneamente. Os resultados mostram o sucesso dos sistemas de detecção e diagnóstico de falhas desenvolvidos, com taxas de sucesso de até 98,78% e 99,47% para as respectivas aplicações UPGN e TEP, com muito poucos padrões confusos e muito bons indicadores de detecção de falhas, como mostrado nas Tabelas 3.5, 3.6, 3.12, 3.13 e 3.15.

Os sistemas propostos foram testados com um estudo de caso univariável e representativo de um problema real (UPGN) e um estudo de caso multivariável com dados simulados (TEP). Com os resultados obtidos para os dois estudos de casos, conclui-se que as RNSP apresentaram uma boa adaptação para o problema de detecção e diagnóstico de falhas em sistemas dinâmicos univariáveis e multivariáveis. A boa adaptação só foi possível mediante a contribuição: da seleção de atributos (variáveis) por classe; do pré-processamento dos dados com os modelos de mapeamento simples e temporal; uso da técnica de *bleaching* para amenizar o uso da aleatoriedade nas saídas das WNN usadas; e dos filtros de agrupamentos usados para a obtenção de diagnósticos por grupos de falhas, vez que a WNN usada realiza a detecção e o diagnóstico de forma simultânea com o diagnóstico por falha.

O modelo de mapeamento temporal com uso das distâncias de Hamming destacou-se no estudo de caso UPGN e o modelo de mapeamento simples destacou-se para o estudo de caso TEP. Com a definição do modelo de mapeamento simples para o TEP e diante da complexidade desta aplicação, a robustez das RNSP para o reconhecimento e classificação de padrões é mostrada neste trabalho. O modelo de mapeamento simples apresenta os padrões individuais para a rede neural. Assim, a recorrência a contribuições passadas acontece apenas com a RNSP e, mais especificamente, no modo de armazenamento dos conteúdos de memórias nas RAM, na fidelidade à metodologia de apresentação dos padrões de entradas para a rede e na técnica de *bleaching* utilizada para a definição do discriminador representante do padrão de teste.

Na rede WiSARD, cada discriminador funciona como um classificador de padrões de uma única classe, o que permite afirmar que a detecção e o diagnóstico de falhas são realizados simultaneamente, aumentando assim a eficiência computacional e flexibilidade da estrutura para a detecção e o diagnóstico de falhas. Esse procedimento difere da maioria dos modelos propostos para o problema de FDD, onde uma estrutura é usada para a detecção de falhas e, posteriormente, outra estrutura é usada para o diagnóstico das falhas detectadas. Também, nas RNSP o treinamento e a teste são realizados sem a necessidade de retreinamento do modelo, conferido assim uma consistência nos resultados obtidos.

Finalmente, o sistema FDD-WiSARD é um método baseado em dados e, portanto, é usado para detectar e diagnosticar falhas que ocorreram no passado. Trabalhos futuros podem melhorar o modelo proposto para avaliar, detectar e classificar falhas novas, desconhecidas ou múltiplas no processo.

## Abreviaturas

ACC:	<i>ACC</i> uracy.
ADL:	Análise Discriminante Linear.
ADQ:	Análise Discriminante Quadrática.
AG:	Algoritmo Genético.
ANFIS:	<i>Adaptive Network-based Inference System</i> .
APS:	Algoritmo de Projeções Sucessivas.
BA:	Estado da Bahia – Brasil.
BM:	<i>bookmaker informedness</i> .
CAPES:	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.
CN:	Condição Negativa.
CNPq:	Conselho Nacional de Desenvolvimento Científico e Tecnológico.
CP:	Condição Positiva.

CPU:	<i>Central Processing Unit.</i>
<i>Disc<sub>k</sub></i> :	Discriminador <i>k</i> .
FDD:	<i>Fault Detection and Diagnosis.</i>
FDD- WiSARD:	Sistema de detecção e diagnóstico de falhas com base na rede WiSARD.
FDI:	<i>Fault Detection and Isolation.</i>
FN:	Falso Negativo.
FP:	Falso Positivo.
GB:	<i>GigaByte.</i>
kNN:	<i>k-Nearest Neighbour.</i>
LR-:	Razão de Verossimilhança Negativa.
LR+:	Razão de verossimilhança positiva.
MCC:	<i>Matthews correlation coefficient.</i>
MK:	<i>Markedness.</i>
MSPCA:	<i>Multi-Scale Principal Component Analysis.</i>
NA:	Número total de atributos usados no conjunto de dados.
NC:	Número de classes.
<i>Offset</i> :	Desvio sistemático.
P:	Prevalência.
Petrobras:	Petróleo Brasileiro S.A.
PIMS:	<i>Plant Information Management System.</i>
PN:	Preditivo Negativo.
PP:	Preditivo Positivo.
PT:	População Total.
<i>Pun<sub>i</sub></i> :	Punição para o atributo <i>i</i> .
RAM:	<i>Random Access Memories.</i>
<i>Rec<sub>i</sub></i> :	Recompensa para o atributo <i>i</i> .
RecPun	RECompensa/ PUNição.
RNA:	Rede Neural Artificial.
RNSP:	Rede Neural Sem Pesos.
<i>RP</i> :	Recompensa/Punição.
SES:	Suavização Exponencial Simples.
SGC:	<i>Sale Gas Compressor.</i>
SPCA <sub>ms</sub> :	Fator de Similaridade PCA <i>Multiscale.</i>
TAH:	<i>Temperature Alarm High.</i>
TAHH:	<i>Temperature Alarm High High.</i>
TDO:	Taxa de Diagnóstico de Probabilidades.
TEP:	<i>Tennessee Eastman Process.</i>
TFD:	Taxa de Falsa Descoberta.
TFN:	Taxa de Falso Negativo.
TFO:	Taxa de Falsa Omissão.

TFP:	Taxa de Falso Positivo.
TVN:	Taxa de Valor Negativo.
TVP:	Taxa de Valor Positivo.
UFBA:	Universidade Federal da Bahia.
UPGN:	Unidade de Processamento de Gás Natural.
VN:	Verdadeiro Negativo.
VNP:	Valor Negativo Preditivo.
VP:	Verdadeiro Positivo.
VPP:	Valor Positivo Preditivo.
WC:	WiSARD Contador.
WiSARD:	<i>Wilkes-Stonham-Aleksander Recognition Device</i> .
WP:	WiSARD Padrão.

## Notações

$\bar{a}_i$ :	Atributo médio dos $t$ valores da série processada para o atributo $i$ .
$d_{a_i,t}$ :	Desvio do atributo $i$ em relação ao padrão de entrada atual $x_t$ .
$2^p$ :	Quantidade de locações de memória endereçadas por $p$ bits.
A(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
$a_h$ :	Atributo da série original no período $h$ .
$a_{n,t}$ :	Atributo $n$ do padrão de entrada $x_t$ .
$a_t$ :	Atributo de entrada no instante $t$ .
B(g):	Inerte gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
C(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
C[a]:	Posição de memória acessada pelo vetor de entradas de entrada I=a.
$C_w$ :	Nível de confiança da rede WiSARD para o reconhecimento de um padrão $w$ .
D(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
$d$ :	Terminal contendo o valor da saída desejada.
D1:	Estado com desvios sistemáticos em relação ao valor normal de operação.
D2:	Estado com desvios sistemáticos em relação ao valor normal de operação.
E(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
$E_h$ :	Valor da série ajustada no período $h$ .
$E_{h-1}$ :	Valor da série ajustada no período $h - 1$ .
F(g):	Subproduto gasoso gerado no simulador <i>Tennessee Eastman Process</i> .
$f(x)$ :	Função que define aproximadamente os limites para as classes consideradas.
$F_j$ :	Padrão de Falha do Tipo $j$ .
G:	Produto de saída no simulador <i>Tennessee Eastman Process</i> .
$h$ :	Horizonte da janela temporal processada.
H:	Produto de saída no simulador <i>Tennessee Eastman Process</i> .
I:	Vetor de entradas para as RAM.
$k$ :	Quantidade de discriminadores na rede WiSARD.

kg/h:	Quilograma por hora.
kPa:	Unidade de medida internacional Quilo Pascal.
kscmh:	Mil unidades de metros cúbicos por hora.
kW:	Quilowatt é uma unidade de potência correspondente a 10000 watts.
m <sup>3</sup> /h:	Metro cúbico por hora.
mol%:	Unidade utilizada para a grandeza quantidade de substância ou matéria.
$n$ :	Dimensionalidade dos vetores de entrada para a aplicação utilizada.
$N$ :	Estado de operação normal do sensor.
$n_i(D_{hf})$ :	Treinamentos no discriminador da classe com maior número de padrões.
$p$ :	Número de <i>bits</i> do vetor de entradas para as RAM.
$P_{2,max}$ :	Segunda maior pontuação obtida por um dos discriminadores da rede WiSARD.
$P_{max}$ :	Pontuação máxima obtida por um dos discriminadores da rede WiSARD.
$q$ :	Define a fase em que o nodo RAM se encontra: aprendizagem ou teste.
$r$ :	Numero de RAM em um discrimindor.
$R1$ :	Instrumento rompido com leituras para o topo de escala (200 °C).
$R2$ :	Instrumento rompido com oscilação binária das leituras entre 0 ° e 200 °C.
$R3$ :	Instrumento rompido com leituras reduzidas para o fundo de escala (0 °C).
$R4$ :	Instrumento rompido parcialmente com leituras oscilando em torno do valor
$R5$ :	Instrumento rompido com oscilação em torno de valores próximos a 0 °C.
$R6$ :	Instrumento rompido com oscilação em torno de valores próximos a 200 °C.
$R_{i,c}$ :	Posição do atributo $i$ na classe $c$ .
$RP_i$ :	Recompensa e Punição para o atributo $i$ .
$V_{i,c}$ :	Valor que representa a importância do atributo $i$ para a classe $c$ .
$W$ :	Coefficiente de ajuste, pertencente ao intervalo $0 < W < 1$ .
$w$ :	Padrão de entradas para a rede neural sem pesos.
$x$ :	Padrão de entrada para o sistema.
$x_t$ :	Padrão de entrada $x$ no instante $t$ .
$y$ :	Saída da rede neural sem pesos com valor igual a um rótulo de classe.

## Símbolos Gregos

$\sigma$ :	Desvio padrão dos $t$ valores da série temporal processada.
$\mu$ :	Média aritmética dos $t$ valores da série temporal processada.
$\pi$ :	Número pi, cujo valor aproximado é 3,1415.
$\Delta_{a_{i,t}}$ :	Grau de pertinência do atributo $i$ para o padrão de entrada $x_t$ .
:	Possíveis valores para o <i>bleaching</i> convencional.
:	Operador de somatório de valores definidos.

## Referências

- Aleksander, I. (1967). Adaptive systems of logic networks and binary memories. *Proceedings of the Spring Joint Computer Conference*, ACM, 30, 707-712.
- Aleksander, I., De Gregorio, M., França, F. M. G., Lima, P. M. V., & Morton, H. (2009). A brief introduction to Weightless Neural Systems. *In ESANN*, 299-305.
- Aleksander, I., Thomas, W. V., Bowden, P. A. (1984). WISARD: a radical step forward in image recognition. *Sensor Review*, 4(3), 120-124.
- Andrade, V. E. (2012). Sistema de detecção e diagnóstico de falhas em sensores de um compressor de gás natural utilizando lógica fuzzy tipo-2. *Dissertação de Mestrado*, Programa de Engenharia Industrial. Universidade Federal da Bahia-UFBA, Salvador, Bahia, Brasil.
- Andrade, V. E., Fontes, C. H., Embiruçu, M. (2011). An interval type-2 fuzzy logic approach for instrument fault detection and diagnosis. *Proceedings of the World Congress on Engineering*, 2, 1008-1012.
- Angelov, P., Yager, R. (2012). A new type of simplified fuzzy rule-based system. *In: International Journal of General Systems*, 41(2), 163-185.
- Araújo, L. A. (2011). RWISARD: um modelo de rede neural sem peso para reconhecimento e classificação de imagens em escala de cinza. *Dissertação de Mestrado*, Programa de Pós-Graduação em Engenharia de Sistemas e Computação. Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.
- Bandeira, L. C., França, H. L., França, F. M. G. (2009). NC-WiSARD: Uma interpretação Booleana da arquitetura neocognitron. *In: Anais do IX Congresso Brasileiro de Redes Neurais e Inteligência Computacional*, CD-Rom, Ouro Preto, Minas Gerais, Brasil, 25-28.
- Barragan, J. F., Fontes, C. H., Embiruçu, M. (2016). A wavelet-based clustering of multivariate time series using a Multiscale SPCA approach. *Computers & Industrial Engineering*, 95, 144-155.
- Barty, M., Patton, R., Syfert, M., de las Heras, S., Quevedo, J. (2006). Introduction to the DAMADICS actuator FDI benchmark study. *Control Engineering Practice*, 14(6), 577-596.
- Beckerle, P., Schaede, H., Butzek, N., Rinderknecht, S. (2012). Balancing filters: an approach to improve model-based fault diagnosis based on parity equations. *Mechanical Systems and Signal Processing*, 29, 137-147.
- Berleant, D. (1991). Combining qualitative and quantitative simulation: in brief. *Proceedings of the Second Annual Conference on AI, Simulation and Planning in High Autonomy Systems. Integrating Qualitative and Quantitative System Knowledge*, IEEE, 233-240.
- Blázquez, L. F., Miguel, L. J. (2005). Additive fault detection in nonlinear dynamic systems with saturation. *ISA Transactions*, 44(4), 515-538.
- Bledsoe, W. W., Browning, I. (1959). Pattern recognition and reading by machine. *Proceedings Eastern Joint Computer Conference*, 225-232.
- Blesa, J., Jiménez, P., Rotondo, D., Nejjari, F., Puig, V. (2014). Fault diagnosis of a wind farm using interval parity equations. *IFAC Proceedings Volumes*, 47(3), 4322-4327.
- Bocaniala, C. D., da Costa, J. S. (2006). Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem. *Control Engineering Practice*, 14(6), 653-669.



- Burns, C. M. (2006). Towards proactive monitoring in the petrochemical industry. *Safety Science*, 44(1), 27-36.
- Cardoso, D. D. O., Lima, P. M., de Gregório, M., Gama, J., França, F. M. (2011). Clustering data streams with weightless neural networks. *European Symposium on Artificial Neural Networks, ESANN2011, Computational Intelligence and Machine Learning*, Bruges, Belgium.
- Cardoso, D., De Gregorio, M., Lima, P., Gama, J., & França, F. (2012). A weightless neural network-based approach for stream data clustering. *International Conference on Intelligent Data Engineering and Automated Learning*, Springer Berlin Heidelberg, 328-335.
- Carvalho, D. S., Carneiro, H. C., França, F. M., Lima, P. M. (2013). B-bleaching: agile overtraining avoidance in the WiSARD weightless neural classifier. *European Symposium on Artificial Neural Networks, ESANN2013, Computational Intelligence and Machine Learning*, 515-520.
- Chetouani, Y. (2008). Design of a multi-model observer-based estimator for fault detection and isolation (FDI) strategy: application to a chemical reactor. *Brazilian Journal of Chemical Engineering*, 25, 777-788.
- Chiang, L. H., Russel, E. L., Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. 1. Edição, Springer.
- Conti, V., Militello, C., Vitabile, S., Sorbello, F. (2009). An embedded fingerprints classification system based on weightless neural networks. *Frontiers in Artificial Intelligence and Application. Proceedings of the 18th Conference on New Directions in Neural Networks: Workshop Italian on Neural Networks, WINN2009*, 193, 67-75.
- De Gregorio, M., Giordano, M. (2014). Change detection with weightless neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 403-407.
- De Souza, A. F., Freitas, F. D., De Almeida, A. G. C. (2010). High performance prediction of stock returns with VG-RAM weightless neural networks. *Workshop on High Performance Computational Finance, WHPCF2010, IEEE*, 1-8.
- Deng, S., Lin, S.-Y., Chang, W.-L. (2011). Application of multiclass support vector machines for fault diagnosis of field air defense gun. *Expert Systems with Applications*, 38(5), 6007-6013.
- Downs, J. J., Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3), 245-255.
- Duan, L., Xie, M., Bai, T., Wang, J. (2016). A new support vector data description method for machinery fault diagnosis with unbalanced datasets. *Expert Systems with Applications*, 64, 239-246.
- Fan, C. F., Yih, S., Tseng, W. H., Chen, W. C. (2013). Empirical analysis of software-induced failure events in the nuclear industry. *Safety Science*, 57, 118-128.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874.
- Grieco, B. P., Lima, P. M., De Gregorio, M., França, F. M. (2010). Producing pattern examples from mental images. *Neurocomputing*, 73(7), 1057-1064.
- Guilford, J. P. (1950). *Fundamental statistics in psychology and education*. 4<sup>a</sup> Edição. New York: McGraw-hill Book.

- Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press, Princeton.
- Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2), 147-160.
- Hell, M., Costa Jr, P., Gomide, F. (2008). Hybrid neurofuzzy computing with nullneurons. *International Joint Conference on Neural Networks, IJCNN2008, World Congress on Computational Intelligence, IEEE*, 3653-3659.
- Herzog, M. A., Marwala, T., Heyns, P. S. (2009). Machine and component residual life estimation through the application of neural networks. *Reliability Engineering System Safety*, 94(2), 479-489.
- Hu, J., Yi, Y. (2016). A two-level intelligent alarm management framework for process safety. *Safety Science*, 82, 432-444.
- Ji, Z., Xia, Q., Meng, G. (2015). A review of parameter learning methods in Bayesian network. In advanced intelligent computing theories and applications. *Springer International Publishing of the series Lecture Notes in Computer Science*, 9227, 3-12.
- Jiang, Q., Yan, X., Zhao, W. (2013). Fault detection and diagnosis in chemical processes using sensitive principal component analysis. *Industrial Engineering Chemistry Research*, 52(4), 1635-1644.
- Johansson, A., Bask, M., Norlander, T. (2006). Dynamic threshold generators for robust fault detection in linear systems with parameter uncertainty. *Automatica*, 42(7), 1095-1106.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of basic Engineering*, 82(1), 35-45.
- Kuncheva, L. I., Whitaker, C. J. (2003). Examining the relationship between majority vote accuracy and diversity in bagging and boosting. *Technical Report School of Informatics, University of Wales, Bangor*.
- Lau, C. K., Ghosh, K., Hussain, M. A., Hassan, C. C. (2013). Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS. *Chemometrics and Intelligent Laboratory Systems*, 120, 1-14.
- Leite, D., Costa, P., Gomide, F. (2012). Evolving granular neural network for fuzzy time series forecasting. *The 2012 International Joint Conference on Neural Networks, IEEE*, 1-8.
- Li, H., Yu, J., Hilton, C., Liu, H. (2013). Adaptive sliding-mode control for nonlinear active suspension vehicle systems using T-S fuzzy approach. *IEEE Transactions on Industrial Electronics*, 60(8), 3328-3338.
- Liao, Z., Wang, B., Xia, X., Hannam, P. M. (2012). Environmental emergency decision support system based on artificial neural network. *Safety Science*, 50(1), 150-163.
- Lo, C. H., Fung, E. H., Wong, Y. K. (2009). Intelligent automatic fault detection for actuator failures in aircraft. *IEEE Transactions on Industrial Informatics*, 5(1), 50-55.
- Ludermir, T. B., Carvalho, A. C. P. L. F., Braga, A. P., de Souto, M. C. P. (1999). Weightless neural models: a review of current and past works. *Neural Computing Surveys*, 2, 41-61.
- Ludermir, T. B., de Souto, M. C., de Oliveira, W. R. (2008). Weightless neural networks: knowledge-based inference system. In *Neural Networks, 2008. SBRN'08. 10th Brazilian Symposium, IEEE*, 207-212.

- Luo, H., Wang, Y., Cui, J. (2011). A SVDD approach of fuzzy classification for analog circuit fault diagnosis with FWT as preprocessor. *Expert Systems with Applications*, 38(8), 10554-10561.
- Ma, M. D., Wong, D. S. H., Jang, S. S., Tseng, S. T. (2010). Fault detection based on statistical multivariate analysis and microarray visualization. *IEEE Transactions on Industrial Informatics*, 6(1), 18-24.
- Maurya, M. R., Rengaswamy, R., Venkatasubramanian, V. (2007). Fault diagnosis using dynamic trend analysis: a review and recent developments. *Engineering Applications of Artificial Intelligence*, 20(2), 133-146.
- McElroy, B., Howells, G. (2011). Automated adaptation of input and output data for a weightless artificial neural network. *International Journal of Database Theory and Application*, 4(3), 49-58.
- Mpofu, N. (2006). Forecasting stock prices using a weightless neural network. *Journal of Sustainable Development in Africa*, 8(1), 115-119.
- Nguyen, N. T., Lee, H. H. (2008). Decision tree with optimal feature selection for bearing fault detection. *Journal of Power Electronics*, 8(1), 101-107.
- Niaki, S. T. A., Abbasi, B. (2005). Fault diagnosis in multivariate control charts using artificial neural networks. *Quality and Reliability Engineering International*, 21(8), 825-840.
- Nurmaini, S., Hashim, S. Z. M., Jawawi, D. N. A. (2009). Modular weightless neural network architecture for intelligent navigation. *International Journal of Advanced in Soft Computing and its Application*, 1(1), 1-18.
- Özyurt, B., Kandel, A. (1996). A hybrid hierarchical neural network-fuzzy expert system approach to chemical process fault diagnosis. *Fuzzy Sets and Systems*, 83(1), 11-25.
- Park, J., Kwon, I.-H., Kim, S.-S., Baek, J.-G. (2011). Spline regression based feature extraction for semiconductor process fault detection using support vector machine. *Expert Systems with Applications*, 38(5), 5711-5718.
- Pouliezos, A., Stavrakakis, G., Lefas, C. (1989). Fault detection using parameter estimation. *Quality and Reliability Engineering International*, 5(4), 283-290.
- Powers, D. M. W. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.
- Rajpal, P. S., Shishodia, K. S., Sekhon, G. S. (2006). An artificial neural network for modeling reliability, availability and maintainability of a repairable system. *Reliability Engineering System Safety*, 91(7), 809-819.
- Ricker, N. L. (1995). Optimal steady-state operation of the Tennessee Eastman challenge process. *Computers and Chemical Engineering*, 19(9), 949-959.
- Rigatos, G., Zhang, Q. (2009). Fuzzy model validation using the local statistical approach. *Fuzzy Sets and Systems*, 160(7), 882-904.
- Romano, D., Kinnaert, M. (2006). Robust design of fault detection and isolation systems. *Quality and Reliability Engineering International*, 22(5), 527-538.
- Santosh, T. V., Vinod, G., Saraf, R. K., Ghosh, A. K., Kushwaha, H. S. (2007). Application of artificial neural networks to nuclear power plant transient diagnosis. *Reliability Engineering System Safety*, 92(10), 1468-1472.

- Saravanan, N., Cholairajan, S., Ramachandran, K. I. (2009). Vibration-based fault diagnosis of spur bevel gear box using fuzzy technique. *Expert Systems with Applications*, 36(2), 3119-3135.
- Sartori, I. (2012). Uma abordagem de desenvolvimento de sistemas de detecção e diagnóstico de falhas baseada em estudos prévios de confiabilidade e sua aplicação ao processamento industrial de gás natural. *Tese de Doutorado*, Programa de Engenharia Industrial. Universidade Federal da Bahia-UFBA, Salvador, Bahia, Brasil.
- Sartori, I., Amaro, C. A., Arduini, F. Z., Souza Júnior, M. B., Embiruçu, M. (2016). Fault detection, diagnosis and correction in industrial processes: a survey of Brazilian research in the global context. *Ciência & Engenharia*, aceito;
- Sartori, I., Amaro, C. A., Souza Júnior, M. B., Embiruçu, M. (2012). Detecção, diagnóstico e correção de falhas: uma proposição consistente de definições e terminologias. *Ciência & Engenharia*, 21(2), 41-53.
- Serdio, F., Lughofer, E., Pichler, K., Buchegger, T., Efendic, H. (2014). Residual-based fault detection using soft computing techniques for condition monitoring at rolling mills. *Information Sciences*, 259, 304-320.
- Sharma, R. S., Dewan, L., Chatterji, S. (2015). Fault diagnosis methods in dynamic systems: review. *International Journal of Electronics and Electrical Engineering*, 3(6), 465-471.
- Silva, G. C., Palhares, R. M., Caminhas, W. M. (2012). Immune inspired fault detection and diagnosis: A fuzzy-based approach of the negative selection algorithm and participatory clustering. *Expert Systems with Applications*, 39(16), 12474-12486.
- Simões Filho, S. (2006). Análise de árvore de falhas considerando incertezas na definição dos eventos básicos. *Tese Doutorado*. Universidade Federal do Rio de Janeiro.
- Simon, D., Simon, D. L. (2010). Analytic confusion matrix bounds for fault detection and isolation using a sum-of-squared-residuals approach. *IEEE Transactions on Reliability*, 59(2), 287-296.
- Sirlantzis, K., Howells, G., Gherman, B. (2009). Novel modular weightless neural architectures for biometrics-based recognition. *Proceedings of the 17th European Symposium on Artificial Neural Networks, ESANN'2009, Advances in Computational Intelligence and Learning*, Bruges, Belgium.
- Soares, A. S., Galvão, R. K. H. (2010). Fault detection using linear discriminant analysis with selection of process variables and time lags. *International Conference on Industrial Technology, ICIT2010*, IEEE, 217-222.
- Souza, C. R., Nobre, F. F., Lima, P. M. V., Silva, R. M., Brindeiro, R. M., França, F. M. G. (2012). Recognition of HIV-1 subtypes and antiretroviral drug resistance using weightless neural networks. *Proceedings of the European Symposium on Artificial Neural Networks, ESANN2012, Computational Intelligence and Machine Learning*, 429-434.
- Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1), 77-89.
- Subhashini, R., Nagarajan, E. (2014). Detection of an incognitos intruder in industries and semantic mapping of emotions. *International Journal of Applied Engineering Research*, 9(20), 6727-6734.
- Ting, K. M. (2011). *Encyclopedia of machine learning*. ISBN 978-0-387-30164-8, Springer.

- Tsymbal, A., Cunningham, P., Pechenizkiy, M., Puuronen, S. (2003). Search strategies for ensemble feature selection in medical diagnostics. *Proceedings of the 16th Symposium on Computer-Based Medical Systems, CBMS2003*, IEEE, 124-129.
- Vale, K. M. O., Neto, A. F., Canuto, A. M. P. (2010). Using a reinforcement-based feature selection method in classifier ensemble. *Proceedings of the 10th International Conference on Hybrid Intelligent Systems, HIS2010*, IEEE, 213-218.
- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N. (2003a). A review of process fault detection and diagnosis: Part II: qualitative models and search strategies. *Computers and Chemical Engineering*, 27(3), 313-326.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S. N. (2003b). A review of process fault detection and diagnosis: Part I: quantitative model-based methods. *Computers and Chemical Engineering*, 27(3), 293-311.
- Wang, W., Li, D. Z., Vrbanek, J. (2012). An evolving neuro-fuzzy technique for system state forecasting. *Neurocomputing*, 87, 111-119.
- Xiao, F., Zhao, Y., Wen, J., Wang, S. (2014). Bayesian network based FDD strategy for variable air volume terminals. *Automation in Construction*, 41, 106-118.
- Yang, S. K., Liu, T. S. (1998). A Petri net approach to early failure detection and isolation for preventive maintenance. *Quality and Reliability Engineering International*, 14(5), 319-330.
- Zadeh, L. A. (2008). Is there a need for fuzzy logic? *Information Sciences*, 178(13), 2751-2779.
- Zakharov, A., Tikkala, V.-M., Jämsä-Jounela, S.-L. (2013). Fault detection and diagnosis approach based on nonlinear parity equations and its application to leakages and blockages in the drying section of a board machine. *Journal of Process Control*, 23(9), 1380-1393.
- Zhang, Y., Zhou, H., Qin, S. J., Chai, T. (2010). Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. *IEEE Transactions on Industrial Informatics*, 6(1), 3-10.

## CHAPTER IV

### Chapter 4: Faults Detection and Diagnosis Based on Granular Weightless Neural Networks and on Selection of Attributes

#### Abstract

This paper presents a fault detection and diagnosis system in multivariate historical data based on *Weightless Neural Networks* (WNN), fuzzy granular computing and selection of attributes. The system starts with a selection of attributes by class, followed by the granulation of the selected attributes. The data granulation is based on interval mathematics and fuzzy logic. The outputs of the granules feed WNN composed of neurons based on RAM (*Random Access Memories*) memory devices. These networks present a high potential for pattern recognition and classification with fast and flexible learning algorithms, accuracy and consistency in the results, without the need of waste generation and retraining. The FDD (*Fault Detection and Diagnosis*) by class or fault clusters are obtained directly from the outputs of the tested networks or through a clustering filter, as a final stage of the FDD system. The proposed approach was tested using simulated/ synthetic data through two benchmark systems used to compare monitoring solutions and assessment of FDD techniques. The first one comprised the well-known *Tennessee Eastman Process* (TEP) and the second comprised a nonisothermal *Continuous Stirred Tank Reactor* (CSTR) operating in closed-loop with temperature and level control. The results show the success of the system for the fault detection and diagnosis with rates of correct classification equal to 99.85% and 99.70% in the TEP and CSTR applications, respectively.

**Keywords:** Fault Detection and Diagnosis; Weightless Neural Networks; Selection of Attributes; Granular Computing; Fuzzy Logic.

## CAPÍTULO IV

### Capítulo 4: Detecção e Diagnóstico de Falhas Baseados em Redes Neurais Sem Pesos Granulares e Seleção de Atributos

#### Resumo

Este capítulo apresenta um sistema de detecção e diagnóstico de falhas em dados históricos multivariáveis baseado em redes neurais sem pesos, computação granular *fuzzy* e seleção de atributos. O sistema inicia com a seleção de atributos por classe, seguida da granulação dos atributos selecionados. A granulação dos dados é fundamentada na matemática intervalar e na lógica *fuzzy*. As saídas dos grânulos criados alimentam redes neurais sem pesos, compostas por neurônios baseados em dispositivos de memórias RAM (*Random Access Memories*, memórias de acesso randômico ou aleatório). Essas redes apresentam um elevado potencial para o reconhecimento e classificação de padrões com algoritmos de aprendizagem rápidos e flexíveis, precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento (Ludermir *et al.*, 1999). A detecção e o diagnóstico de falhas (FDD, *Fault Detection and Diagnosis*) por classe ou grupos de falhas são obtidos diretamente nas saídas das redes testadas ou por um filtro de agrupamento, como etapa final do sistema de FDD. O sistema foi testado com dois estudos de casos baseados em dados simulados. Um deles corresponde a uma simulação de uma planta industrial, bem conhecida na literatura como *Tennessee Eastman Process* (TEP). O outro simula um “reator tanque agitado continuamente” (CSTR, *Continuous Stirred Tank Reactor*). Os resultados mostram o sucesso do sistema para a detecção e o diagnóstico de falhas com percentuais de acertos na classificação de 99,85% e (TEP) e 99,70% (CSTR) para as aplicações.

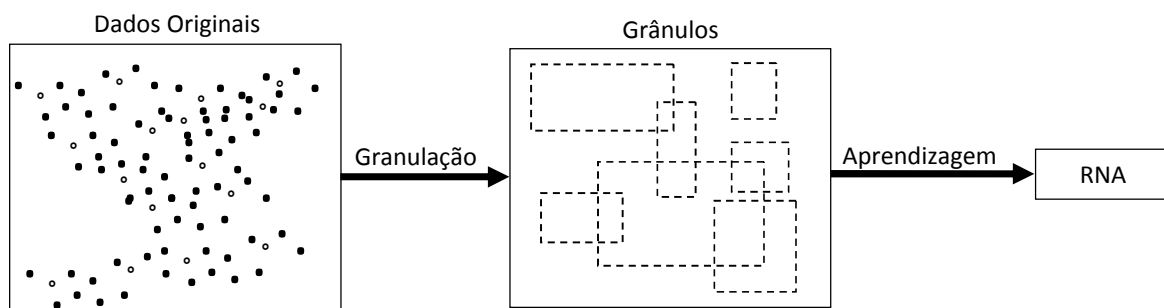
**Palavras-chave:** Detecção e Diagnóstico de Falhas; Redes Neurais sem Pesos; Seleção de Atributos; Computação Granular; Sistema de Inferência *Fuzzy*.

## 4.1 Introdução

A detecção e o diagnóstico de falhas (FDD, *Fault Detection and Diagnosis*) em processos industriais é uma área de estudos de grande interesse da comunidade científica. Dispor de mecanismos para a detecção antecipada e segura de falhas diminui os riscos de mau funcionamento ou paradas não programadas dos sistemas. Conseqüentemente, aumenta-se a confiabilidade nos equipamentos, evitam-se perdas materiais e, até mesmo, acidentes de trabalho e ambientais (Blázquez e Miguel, 2005; Burns, 2006; Chiang *et al.*, 2001; Fan *et al.*, 2013; Romano e Kinnaert, 2006; Yang e Liu, 1998).

Com os avanços tecnológicos nos sistemas de controle e monitoramento de processos industriais aumenta-se cada vez mais a disponibilidade de dados para fins de análises e diagnósticos. Com isso os bancos de dados passam a se configurar como ferramentas importantes para aquisição de conhecimento e desenvolvimento de novos sistemas ou aperfeiçoamento dos já existentes. No entanto, o armazenamento de dados pode ser oneroso e em muitos casos impossível ou impraticável. Por isso, é crescente a busca por sistemas com aprendizado incremental a partir de séries de dados históricos (Hong e Wang, 2014; Liu, 2007), com destaque para os sistemas neurais granulares evolutivos (Angelov e Zhou, 2006; Angelov e Yager, 2011; Chen, 2013; Leite *et al.*, 2009; Leite *et al.*, 2012; Li, 2010; Pedrycz e Gomide, 2007). Nesses sistemas, além dos ajustes nos parâmetros característicos dos sistemas adaptativos, também acontecem adaptações nas suas estruturas com a evolução do processamento dos dados (Angelov e Zhou, 2006; Pradhan *et al.*, 2006). Neste contexto, este trabalho aborda a detecção e diagnóstico de falhas em processos industriais dinâmicos baseados na inteligência computacional (computação granular *fuzzy* e redes neurais sem pesos) e em um método de seleção de atributos por classe. Os sistemas de FDD propostos foram construídos, testados e avaliados usando dados históricos obtidos por simuladores ou plantas virtuais.

Para Leite *et al.* (2009), o princípio da computação granular é atuar em um nível de detalhamento maior e mais abstrato, não se detendo apenas à questão da precisão para a modelagem dos sistemas computacionais. Os sistemas neurais granulares evolutivos foram inicialmente propostos para a aprendizagem incremental *on-line* e classificação de fluxo de dados, sem a necessidade de retreinamento de modelos (Pedrycz e Vukovich, 2001; Pedrycz, *et al.*, 2004). A estrutura de um sistema neural granular evolutivo compreende basicamente as etapas de granulação e aprendizagem, conforme mostrado na Figura 4.1 (Pedrycz e Vukovich, 2001). Na “primeira etapa”, grânulos de informações são construídos a partir de dados numéricos originais e modelos *fuzzy*. Na “segunda etapa”, a aprendizagem na rede neural acontece com base nas informações contidas nos grânulos. Assim, a rede neural não precisa ter acesso a todos os dados de treinamento, mas às informações atualizadas dos grânulos criados.



**Figura 4.1:** Estrutura de um Sistema Neural Granular [adaptado de Leite *et al.* (2009)].

Considerando o aprendizado para classificação ou agrupamento de dados, destacam-se as seguintes características dos sistemas granulares evolutivos neuro-*fuzzy* (Leite *et al.*, 2009): a)



possibilidade de uso de aprendizagem supervisionada ou não supervisionada; b) não é requerido um conhecimento prévio das propriedades estatísticas dos dados ou classes; c) o aprendizado acontece durante o processamento de todo o fluxo de dados em uma única apresentação; d) adaptação da estrutura e dos parâmetros para o “aprendizado” de um novo conceito ou “esquecimento” de informações que não são mais relevantes; e) limites discriminantes entre as classes, definidos com a granulação do espaço de atributos; f) boa adaptação a ambientes dinâmicos e incerteza nos dados. Quando a rede neural considerada evolui também a sua arquitetura, ela é referenciada como “rede neural granular evolutiva”.

No trabalho aqui proposto, o termo evolutivo é relativo ao sistema e não à rede neural usada. A evolução acontece apenas com a granulação dos dados de entrada e nos princípios da lógica *fuzzy*. As redes neurais usadas são consideradas granulares por aprenderem padrões comportamentais definidos a partir dos grânulos criados. Essas redes pertencem a uma classe conhecida como “Redes Neurais Sem Pesos” (RNSP), propostas inicialmente por Aleksander (1967). As RNSP são modelos digitais baseados em dispositivos de memória de acesso randômico ou aleatório (RAM, *Random Access Memories*). O aprendizado nas RNSP acontece em “memórias” inseridas no próprio neurônio, em forma de tabelas-verdade. A viabilidade de uso dessas memórias atribui vantagens às RNSP em relação aos modelos com pesos, tais como: baixo custo operacional; semelhança com os sistemas digitais convencionais; algoritmos de aprendizagem rápidos e flexíveis; precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento das redes; e principalmente o grande potencial das RNSP para o reconhecimento e a classificação de padrões.

Inicialmente os modelos neurais sem pesos eram totalmente booleanos e voltados quase que exclusivamente para o reconhecimento e a classificação de imagens digitais. Mas atualmente as RNSP passaram a ter aplicações diversificadas que vão além do reconhecimento e classificação de padrões, dentre as quais destacam-se: “reconhecimento de dígitos e impressões digitais” (Bandeira *et al.*, 2009; Conti, *et al.*, 2009; Grieco *et al.*, 2010); “reconhecimento de faces e impressões faciais” (Araújo, 2011; Sirlantzis *et al.*, 2009; Subhashini e Nagarajan, 2014); “navegação de robôs” (Nurmaini *et al.*, 2009; McElroy e Howells, 2011); “agrupamento de fluxos de dados” (Cardoso *et al.*, 2011); e “previsão de séries temporais” (De Souza *et al.*, 2010; Mpofu, 2006). No entanto, na literatura relacionada à detecção e diagnóstico de falhas em sistemas dinâmicos, nenhum trabalho foi encontrado com o uso das RNSP. Os estudos apresentados por De Gregorio e Giordano (2014) e Cardoso *et al.* (2012) aproximam do contexto. De Gregorio e Giordano (2014) propuseram um sistema, denominado de CwisarDH, para o problema da detecção de mudanças no campo de visão de uma câmera. O sistema proposto utiliza um discriminador para cada ponto de cobertura do vídeo com o conceito de cores. Cardoso *et al.* (2012) apresentaram o sistema StreamWiSARD para agrupamento de dados de fluxos com janela deslizante. O sistema é composto discriminadores WiSARD como unidades primárias e capaz definir *clusters* de alta qualidade, restrito a um número pequeno de *microclusters*. A ausência de sistemas de FDD baseados nas RNSP, e o potencial dessas redes para o reconhecimento e classificação de padrões, foram os principais fatores que motivaram o desenvolvimento deste trabalho.

Os sistemas desenvolvidos foram validados por dois estudos de caso multivariáveis baseados em dados simulados. Um deles refere-se a uma planta industrial considerada como um *benchmark* na área de detecção e diagnóstico de falhas e conhecida como *Tennessee Eastman Process* (TEP) (Downs e Vogel, 1993; Ricker, 1995). O outro estudo de caso simula um reator tanque agitado continuamente (CSTR, *Continuous Stirred Tank Reactor*) não isotérmico (Singhal e Seborg, 2002).

Na continuidade, a seção 4.2 apresenta os fundamentos sobre redes neurais sem pesos, e a seção 4.3 descreve o sistema FDD-RNSPG (FDD- RNSPG, *Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Granulares*) e a metodologia utilizada para os treinamentos e testes realizados. Na seção 4.4 são apresentados os dois estudos de caso utilizados com os seus respectivos resultados. Por último, na seção 4.5, são apresentadas as conclusões sobre os sistemas FDD-RNSPG propostos para os estudos casos considerados.

## 4.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA), também conhecidas como sistemas conexionistas, computação neural ou processamento paralelo distribuído, são atualmente uma área com fronteiras que vão das engenharias às ciências cognitivas. Inspiradas inicialmente na funcionalidade dos neurônios biológicos, a ideia básica em redes neurais é que pequenas unidades de processamento simples, dispostas em uma ou mais camadas e interagindo umas com as outras e com o ambiente, podem ter elevado poder computacional e capacidade de serem ajustadas ao comportamento desse ambiente. As unidades de processamento recebem o nome de neurônios, também conhecidos como nodos ou neurônios artificiais, e a estrutura formada pelas unidades e as interconexões entre elas recebe o nome de rede neural ou rede neural artificial. Por meio de um processo de treinamento, as redes neurais passam a ser capazes de realizar atividades ditas “inteligentes”, mesmo que os dados utilizados nesse treinamento sejam não lineares, incompletos ou, até mesmo, contraditórios. A habilidade em lidar com incertezas faz com que as redes neurais sejam extremamente eficazes em tarefas onde especialistas não estão à disposição ou um conjunto de regras não pode ser facilmente formulado.

As RNA são treinadas com algoritmos de aprendizagem que correspondem a um conjunto de regras bem definidas para a solução de um determinado problema. Os algoritmos de aprendizagem podem variar de acordo com os diferentes modelos de redes neurais, mas são essencialmente derivados de dois neurônios básicos: o modelo de *McCulloch-Pitts* (MCP) (McCulloch e Pitts, 1943), inspirado na funcionalidade do neurônio natural e conhecido como neurônio “com pesos”, em função da ponderação que ele faz sobre as suas entradas; e o modelo digital, sem nenhuma relação com o neurônio natural, baseado em dispositivos RAM e referenciado como neurônio/nodo “sem pesos” (Aleksander, 1967). Os sistemas compostos por neurônios sem pesos são chamados de “redes neurais sem pesos” (RNSP) e descritos com mais detalhes na seção seguinte.

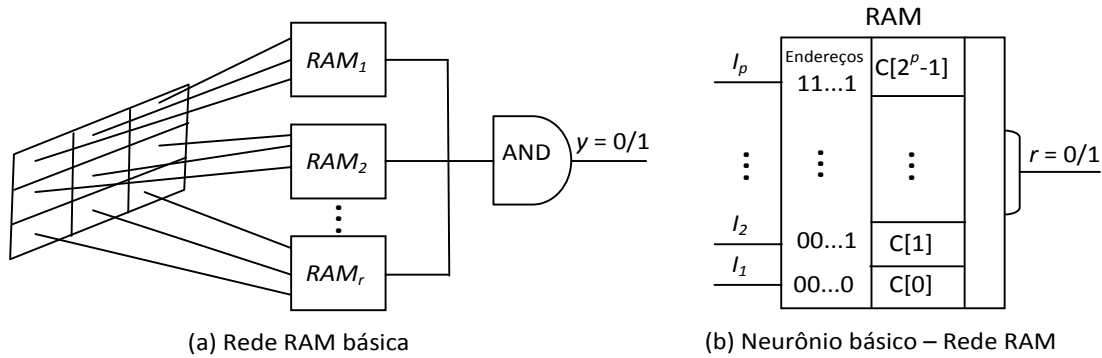
### 4.2.1 Redes Neurais sem Pesos

As RNSP, diferentemente dos modelos convencionais, armazenam as informações em tabelas-verdade, inseridas nos neurônios/nodos baseados em memórias RAM. O aprendizado nessas redes geralmente consiste em mudanças nos conteúdos das tabelas-verdade, o que resulta em algoritmos de aprendizagem rápidos e flexíveis. Na literatura, os termos *RAM-based* e *N-tuple based* também são usados para referir-se às redes neurais sem pesos. As RNSP têm suas origens no método das *n-tuple* de Bledsoe e Browning (Bledsoe e Browning, 1959). O desenvolvimento do nodo digital, baseado em dispositivos de memória de acesso aleatório, foi, sem dúvida, o passo inicial para o desenvolvimento das pesquisas para a computação neural sem pesos.

#### 4.2.1.1 Modelo RAM

O primeiro modelo de RNSP é conhecido como rede RAM, mostrado na Figura 4.2(a). Essa rede é formada por nodos RAM [Figura 4.2(b)] e classifica os padrões de teste como

pertencentes ou não à classe dos padrões treinados (Oliveira *et al.*, 2017). Essa rede pode ser comparada com a rede com pesos do tipo *perceptron* com um único neurônio (Rosenblatt, 1958). O padrão de entrada (atributo qualitativo/quantitativo, univariável/multivariável) binarizado para uma rede RAM é mapeado para as entradas das RAM com base em um modelo de mapeamento pré-definido, conforme mostrado nas Figuras 4.2(a) e 4.3(a). Com o mapeamento usado, um nodo RAM aprende parte dos padrões de entrada para a rede. Com isso, a generalização só acontece em nível da rede, diferentemente do *perceptron* que é capaz de generalizar apenas com um único neurônio.



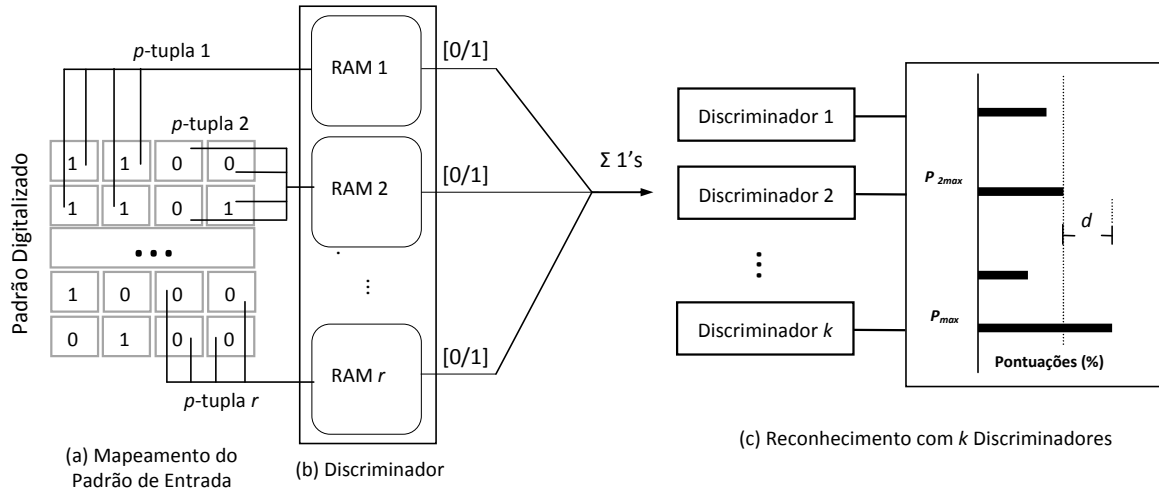
**Figura 4.2:** Estrutura da Rede e do Neurônio RAM (Oliveira *et al.*, 2017).

Um nodo RAM com  $p$  entradas possui  $2^p$  posições de memória. Um padrão binário  $I = \{I_1, I_2, \dots, I_p\}$  nas linhas de entrada acessa apenas uma dessas locações (Ludermir *et al.*, 1999). O conteúdo  $C[a]$  em todas as locações de memórias, sendo  $a$  o endereço acessado pelo padrão  $I = \{I_1, I_2, \dots, I_p\}$ , é inicializado com o valor 0 (zero). O aprendizado na rede RAM consiste em modificar o conteúdo acessado pelas  $p$  linhas de entrada para o valor desejado (0 ou 1). Na fase de teste, o *bit* armazenado na posição  $C[a]$  de memória é a saída do nodo  $r = C[a]$ . Se o valor encontrado na posição de memória  $C[a]$  for 0, significa que o último aprendizado ocorrido foi 0 ou que aquele endereço não foi acessado durante o treinamento. Assim como nas RAM, a saída  $y$  da rede RAM também é 0 ou 1. A saída da rede é igual a 1 somente se as saídas de *todas* as RAM forem iguais a 1. Caso contrário, o padrão será incluído na classe 0 (zero) como padrões não pertencentes à classe de padrões treinados. Desse modo, uma rede RAM básica reconhece somente padrões vistos no treinamento. Porém, se no problema em estudo, duas ou mais classes são requeridas, a estrutura de discriminadores com a rede WiSARD (Wilkie, Stonham e Aleksander's Recognition Device, dispositivo de reconhecimento de Wilkie, Stonham e Aleksander) (Aleksander *et al.*, 1984), descrita na seção seguinte, é utilizada.

#### 4.2.1.2 A Rede WiSARD

A rede neural sem pesos WiSARD possui uma estrutura *feedforward* de uma única camada de discriminadores, voltados para o reconhecimento de padrões (Aleksander *et al.*, 1984). Cada discriminador é composto por  $r$  nodos RAM com  $p$  bits de entrada, 1 (um) bit de saída e é treinado para representar uma única classe de padrões. O padrão de entrada para o discriminador a ser treinado tem todos os seus  $r \times p$  bits conectados, através do mapeamento biunívoco pré-definido, aos bits de entrada das RAM [Figura 4.3(a)]. A quantidade  $r$  de nodos nos discriminadores é inversamente proporcional à quantidade  $p$  de bits nas linhas de entrada das RAM. Quanto mais bits por endereço, menos nodos irão compor os discriminadores. Na fase de teste, o mesmo padrão de entrada é apresentado a todos os discriminadores. Assim, cada RAM identifica o endereço de memória apontado pela  $p$ -tuple correspondente e apresenta como saída o valor armazenado, 0 ou 1, conforme mostrado na Figura 4.3(b). A saída de cada

discriminador corresponde ao número de RAM com saída igual a 1 (1's). O padrão de teste é reconhecido pelo discriminador com “maior” número de RAM com saída igual a 1, de acordo com a Figura 4.3(c). O critério do “maior” e não “todas” como na RAM básica, permite os discriminadores da rede WiSARD reconhecerem padrões não vistos durante o treinamento. A rede WiSARD pode ser comparada a uma a rede *perceptron* com dois ou mais neurônios.



**Figura 4.3:** Representação Esquemática da Rede WiSARD.

Conforme mencionado anteriormente, a quantidade de memória em um nodo RAM é uma função exponencial de suas entradas. Na prática valores elevados de  $p$  não são indicados em função da quantidade elevada de memória requerida e, conseqüentemente, do espalhamento dos endereços acessados. Porém, valores pequenos de  $p$  com conjuntos grandes de padrões de treinamento provocam a saturação dos endereços de memória para os nodos RAM. A saturação acontece quando todos ou quase todos os endereços têm o seu conteúdo alterado para 1 (um). Conseqüentemente as RAM passam a apresentar, com maior frequência, saídas iguais a 1 (um) e empates nas pontuações máximas apresentadas pelos discriminadores. Neste caso a rede WiSARD padrão usa o critério de aleatoriedade para definir um dos discriminadores empatados como vencedor. Porém, o uso da aleatoriedade diminui a acurácia da rede, pois o padrão testado tem grande possibilidade de ser classificado erroneamente.

Para amenizar o problema da saturação dos nodos RAM e do uso da aleatoriedade entre os discriminadores, Aleksander e Morton (1990) propuseram armazenar nos endereços de memória um contador de valor inteiro, ao invés dos valores 0 e 1. Os contadores são inicializados com o valor 0 e incrementados toda vez que um mesmo endereço for acessado durante o treinamento. O uso do contador como conteúdo de memória permitiu o uso da técnica de *bleaching* (refinamento) na fase de teste da rede. A técnica de *bleaching* consiste em usar um valor limiar para realizar um desempate mais confiável entre os discriminadores de uma rede WiSARD (Carvalho *et al.*, 2013). Com essa técnica, a saída de uma RAM é igual a 1 se o valor inteiro armazenado for igual ou superior ao valor do limiar.

Na prática o valor do *bleaching* (limiar) pode ser implementado de duas formas (Carvalho *et al.*, 2013): “homogênea”; ou “heterogênea”. No *bleaching* homogêneo, o mesmo valor de limiar é usado para todos os discriminadores. A cada verificação, se o conteúdo acessado é maior do que o limiar, a saída da RAM é 1. Porém, se dois ou mais discriminadores apresentarem o mesmo somatório de valores 1, o valor do limiar é acrescido de uma unidade. Com isso, os conteúdos das RAM serão novamente verificados de modo a reduzir o número de RAM com saída igual a 1. Este procedimento é repetido até que um discriminador seja eleito ou até que todos parem de pontuar, situação em que a escolha é feita de modo aleatório. No *bleaching*

heterogêneo, o valor do limiar varia de acordo com a quantidade de aprendizado em cada discriminador. Dessa forma, evita-se que uma categoria com mais treinamentos seja “privilegiada”, pois os limiares heterogêneos têm a mesma representatividade e criam um equilíbrio entre os discriminadores (Carvalho *et al.*, 2013). Uma representação gráfica e detalhada do uso do *bleaching* é apresentada em Carneiro *et al.* (2015) e Cardoso *et al.* (2016). A escolha por um dos modelos acontece de forma prática com a realização de testes. Porém, quando o quantitativo de padrões de treinamentos entre as classes consideradas para o estudo de caso difere de forma significativa, espera-se que o *bleaching* heterogêneo apresente resultados mais consistentes.

Antes da WiSARD, uma das dificuldades com as redes RAM referia-se aos conteúdos acessados com valor 0 (zero) na fase de teste. A rede não era capaz de diferenciar se os conteúdos com valor 0 (zero) foram efetivamente aprendidos ou se aqueles endereços foram inicializados com 0 (zero) e não foram acessados durante o treinamento. Para superar essa dificuldade, Kan e Aleksander (1987) propuseram uma versão probabilística do nodo RAM, chamado de PLN (*Probabilistic Logic Neuron*, neurônio lógico probabilístico) e descrito na seção seguinte.

#### 4.2.1.3 Modelo PLN

A principal diferença do neurônio lógico probabilístico PLN em relação ao neurônio RAM é a inclusão de um terceiro valor lógico como conteúdo a ser armazenado nos endereços de memória. Este valor lógico geralmente é denotado por  $u$  e, semanticamente, representa um valor indefinido. Com o valor  $u$  o nodo PLN consegue identificar padrões de testes que não foram apresentados durante o treinamento. Para isso todos os endereços de memória são inicializados com o valor  $u$  e no aprendizado os endereços acessados são alterados para 0 ou 1. Na fase de teste, se um endereço com valor  $u$  é acessado, o nodo PLN produz uma resposta 0 ou 1 com a mesma probabilidade. Assim, a função de saída deixa de ser determinística como no neurônio RAM e torna-se probabilística, com a saída do nodo definida pela Equação 4.1:

$$y = \begin{cases} 0 & \text{se } C[a] = 0 \\ 1 & \text{se } C[a] = 1 \\ \text{random}(0,1) & \text{se } C[a] = u \end{cases} \quad (4.1)$$

com  $C[a]$  sendo o conteúdo acessado pelo padrão de entrada  $a$  e a função aleatória  $\text{random}(0,1)$  produz uma saída 0 ou 1 com a mesma probabilidade. Com a proposição do nodo PLN, outros modelos de neurônios sem pesos surgiram, a exemplo do nodo PLN multivalorado, chamado de MPLN (*Multiple-valued Probabilistic Logic Neuron*) (Myers e Aleksander, 1988, 1989), descrito na seção seguinte.

#### 4.2.1.4 Modelo MPLN

A diferença básica do neurônio lógico probabilístico multivalorado MPLN em relação ao PLN é a viabilidade de armazenar a probabilidade de o neurônio responder com 1 quando um endereço é acessado ao invés de armazenar um elemento do conjunto  $[0, 1, u]$ . No MPLN, os conteúdos de memória são inicializados com 0,5, o equivalente ao valor  $u$  do PLN. Com o processo de aprendizagem, os conteúdos de memória são incrementados ou decrementados, podendo chegar a 0 (limite inferior) ou a 1 (limite superior) (Myers e Aleksander, 1988). A função de ativação para o nodo MPLN aplicada ao conteúdo endereçado é definida pela Equação 4.2:

$$y = \begin{cases} C[a] - \eta g(r) & \text{punição} \\ C[a] + \eta g(r) & \text{recompensa} \end{cases} \quad (4.2)$$

onde  $C[a]$  é o conteúdo de memória endereçado pelo padrão de entrada  $a$ ,  $\eta$  é a taxa de aprendizado e  $g(r)$  definida pela Equação 4.3:

$$g(r) = \begin{cases} +1 & \text{se } r = 1 \\ -1 & \text{se } r = 0 \end{cases} \quad (4.3)$$

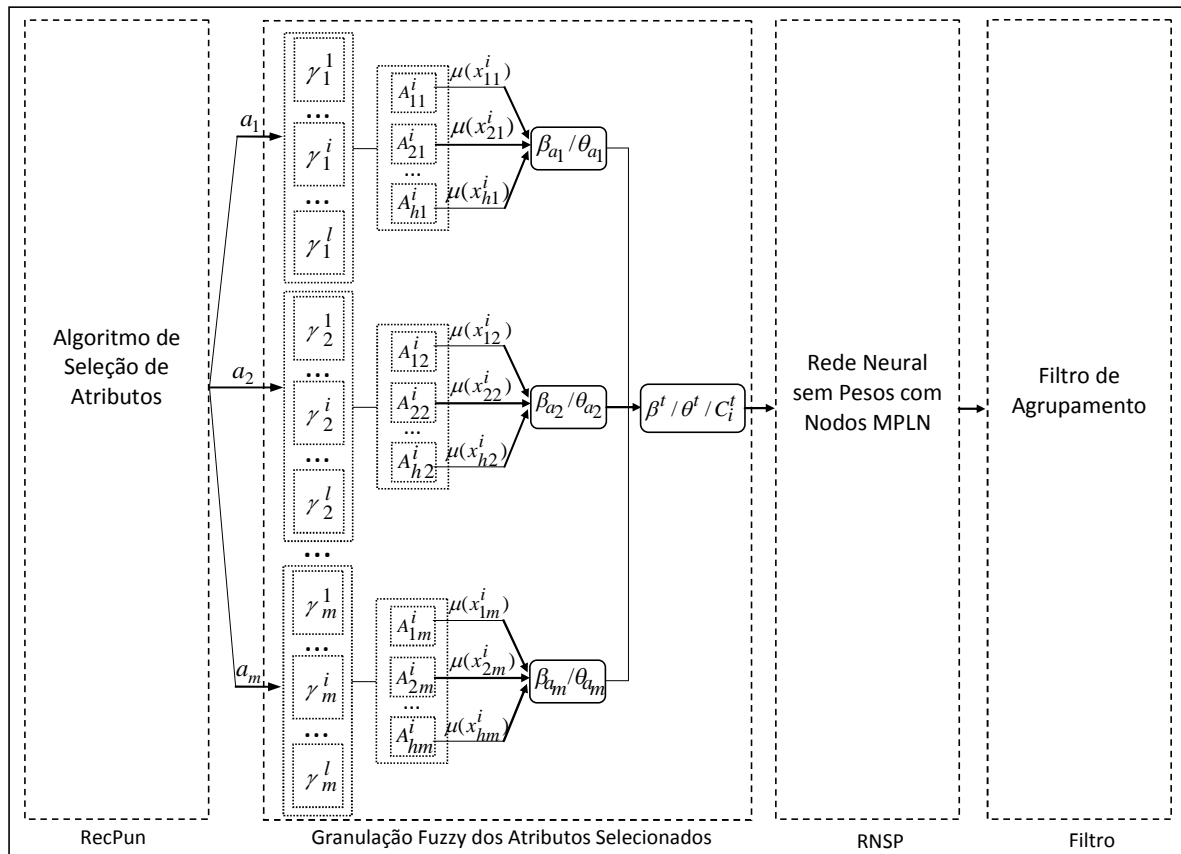
Com  $r$  sendo a saída desejada. Um dos parâmetros que influencia a rapidez com que cada valor chegue a 0 ou a 1 é o número de elementos definidos no conjunto de possíveis valores armazenáveis. Como exemplo, ao usar o conjunto  $[0; 0,5; 1]$  (como no PLN) e o conjunto  $[0; 0,2; 0,4; 0,6; 0,8; 1]$ , é esperado que os conteúdos  $[0; 0,5; 1]$  cheguem primeiro aos extremos 0 e 1. A maneira como o conteúdo inicial é modificado para chegar a 0 ou 1 consiste na estratégia de punição ou recompensa usada na fase de aprendizagem. Desse modo, uma simples punição ou recompensa não é suficiente para definir ou apagar completamente um comportamento, pois a mudança ocorre de forma gradativa e mediante os vários exemplos de treinamento (Myers e Aleksander, 1988).

Um cuidado que se deve ter com as redes compostas por nodos MPLN é com o número de *bits* utilizados para a representação das probabilidades de disparo dos neurônios para gerar uma saída 1. Se esse número for muito grande, a convergência implica num período de tempo maior e num aprendizado mais lento para um conjunto de treinamento com muitos exemplos. Sendo assim, o intervalo de tempo necessário para que as probabilidades armazenadas nas locações de memória sejam atualizadas por punições ou recompensas depende dessa quantidade de *bits* utilizados. Entretanto, uma boa estimativa no número de *bits* para a representação dessas probabilidades depende da especificidade do problema.

### 4.3 O Sistema de Detecção e Diagnóstico de Falhas FDD-RNSPG Proposto

#### 4.3.1 Estrutura do Sistema

O sistema proposto neste trabalho realiza a detecção e o diagnóstico de falhas em dados históricos multivariáveis (vetores  $n$ -dimensionais). Cada dimensão corresponde a uma variável (atributo) obtida aqui por um simulador. Em termos estruturais, o sistema é composto por quatro etapas distintas e complementares, denominadas de (Figura 4.4): “algoritmo de seleção de atributos”; “granulação *fuzzy* dos atributos selecionados”; “rede neural sem pesos”; e “filtro de agrupamentos”. Inicialmente o sistema de FDD utiliza o algoritmo de seleção de atributos por classe proposto por Vale *et al.* (2010) com algumas modificações. A granulação dos atributos selecionados é realizada por um algoritmo de agrupamento não supervisionado apresentado por Leite *et al.* (2009) e adaptado aqui para lidar com dados multivariáveis. As redes neurais sem pesos testadas realizam a detecção e o diagnóstico de falhas de forma paralela ou em cascata e são constituídas por nodos RAM-based do tipo MPLN. A detecção e o diagnóstico (por tipo ou grupo de falhas) de falhas podem ser obtidos diretamente das redes ou por um filtro de agrupamento como etapa final do sistema.



**Figura 4.4:** Estrutura do Sistema Neural sem Pesos: FDD-RNSPG.

No sistema da Figura 4.4, a detecção e o diagnóstico de falhas acontece por meio do reconhecimento e classificação de padrões, definidos por uma aproximação para a relação  $y = f(x)$ , sendo  $x$  um vetor  $n$ -dimensional ( $n$  atributos) de valores reais e  $y$  uma variável de resposta. Nessa relação  $y$  é um rótulo de classe com valor no conjunto  $\{C_1, C_2, \dots, C_k\}$ , com  $k \in \mathbb{N}$  e a função  $f(x)$  define, de forma aproximada, os limites para as classes consideradas. Num contexto temporal, e considerando apenas a detecção de falhas, os atributos  $a_1^t, a_2^t, \dots, a_n^t$  deverão conter informações necessárias para classificar o padrão  $x = [a_1^t, a_2^t, \dots, a_n^t]$  como estado de operação “normal” ou estado de “falha” no instante  $t$ . A partição do estado de falha em subclasses com rótulos definidos pelos diferentes tipos ou grupos de falhas considerados compreende o diagnóstico de falhas realizado pelo sistema. Para melhor compreensão, as etapas do sistema FDD-RNSPG são apresentadas com mais detalhes nas seções 4.3.2 a 4.3.5.

### 4.3.2 Seleção de Atributos

A estrutura utilizada para realizar a detecção e o diagnóstico de falhas neste trabalho são as redes neurais sem pesos com multiclassificadores, treinados individualmente para representar uma única classe de padrões. No entanto, realizar a detecção e o diagnóstico de falhas para um conjunto de padrões  $n$ -dimensionais, considerando os  $n$  atributos para todos os classificadores, é uma tarefa não trivial e com dificuldades naturais para a definição dos limites entre as classes consideradas. Porém, uma boa partição do espaço de estados exige que os classificadores envolvidos sejam “distintos” nas generalizações, de modo a tornar o conjunto de classificadores um sistema “preciso” e “eficaz” no reconhecimento e na classificação de padrões. Para isso, a seleção de atributos em aplicações multivariáveis contribui significativamente para uma

distinção mais precisa entre os classificadores do sistema (Kuncheva e Whitaker, 2003). Uma diferenciação dos atributos para os classificadores envolvidos possibilita a distinção desejada nas generalizações (Tsymbal *et al.*, 2003).

Com base nessas considerações, o presente trabalho utiliza o algoritmo de seleção de atributos por classe denominado de RecPun (RECompensa/ PUNição). Esse algoritmo foi proposto por Vale *et al.* (2010) e baseia-se na recompensa e punição para ordenar os atributos de acordo com as suas importâncias para cada uma das classes existentes. A ideia do modelo é que exista, no mínimo, um classificador por classe e que os atributos selecionados para uma dada classe sejam importantes somente para a referida classe.

A cada padrão processado, o algoritmo RecPun seleciona, de forma não excludente, uma quantidade fixa ( $Q$ ) de atributos em três etapas, a saber: (a) os atributos são ordenados por classe de acordo com a variância ou a correlação de Spearman (Guilford, 1950); (b) calcula-se a recompensa e punição dos atributos ordenados, aqui referenciados pela variância; (c) os atributos são reordenados com base no parâmetro  $RP$  (Recompensa/Punição) definido como base nas Equações 4.4 a 4.6 seguintes (Vale *et al.*, 2010):

$$RP_i = Rec_i - Pun_i \quad (4.4)$$

$$Rec_i = V_{i,c} + \frac{QA}{QA + R_{i,c}} \quad (4.5)$$

$$Pun_i = \frac{1}{QC - 1} \cdot \sum_{c=1}^{QC, c \neq i} \left( V_{i,c} + \frac{QA}{QA + R_{i,c}} \right) \quad (4.6)$$

onde  $QC$  é a quantidade de classes,  $V_{i,c}$  é o valor que representa a importância do atributo  $i$  para a classe  $c$ ,  $R_{i,c}$  é a posição do atributo  $i$  na classe  $c$ , e  $QA$  é a quantidade de atributos usados no conjunto de dados.

Porém, ao aplicar o algoritmo RecPun aos estudos de caso considerados neste trabalho, os atributos selecionados foram os mesmos para a maioria das classes. Para amenizar a situação e promover uma maior diferenciação entre os atributos selecionados por classes, foi adicionada uma etapa final (d) ao algoritmo RecPun, de modo que a relação biunívoca entre as classes permita no máximo 50% de atributos comuns (método aqui denominado de RecPun modificado). Para isso, os atributos selecionados na etapa (c) para uma classe  $i$ , com  $i = 1, 2, \dots, k$ , são comparados com os atributos das classes anteriores. Nessa comparação biunívoca, se a quantidade de atributos comuns for superior a 50%, mantêm-se os primeiros 50% e os demais são substituídos por atributos de menor importância. Porém, se na aplicação considerada essa correspondência biunívoca não for possível, devido ao número reduzido de atributos e um número maior de classes, os atributos comuns são considerados numa ordenação diferente. Essa diferenciação na ordenação dos atributos também contribui para a definição dos limites entre as classes consideradas.

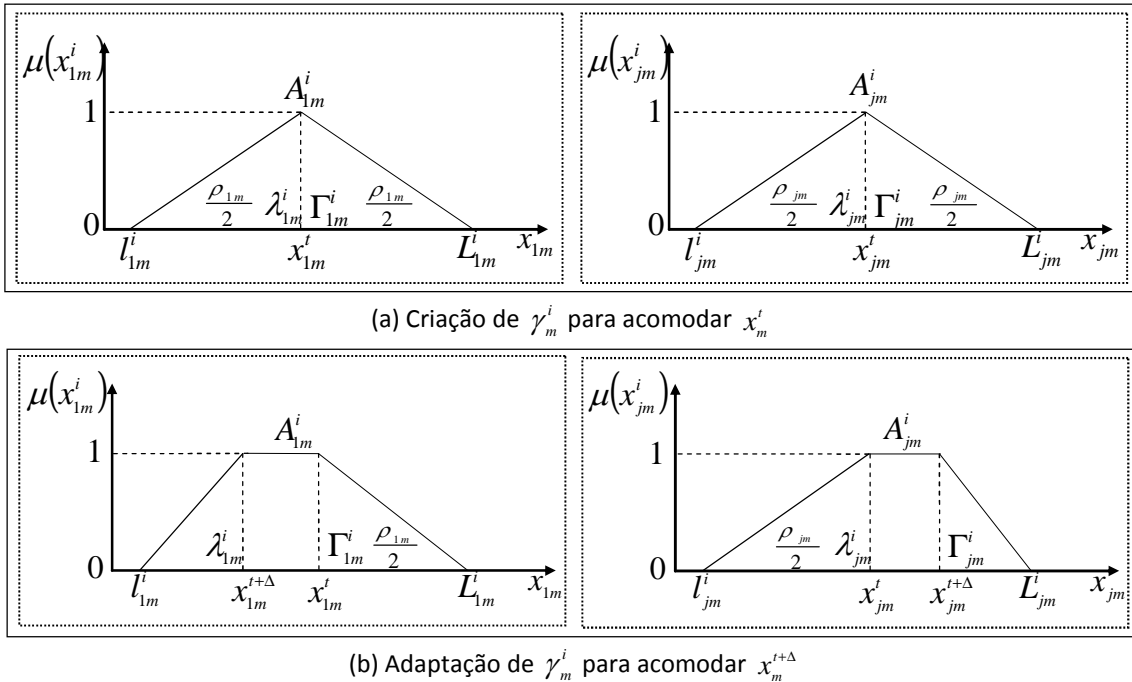
### 4.3.3 Granulação *Fuzzy* dos Atributos Selecionados

A granulação cria uma partição do espaço de entrada com a formação evolutiva de grânulos de informação extraídos do fluxo de dados. Neste trabalho os grânulos são criados tanto na etapa de aprendizagem quanto de teste e consiste numa adaptação do algoritmo apresentado por Leite *et al.* (2009). Conforme mostrado na Figura 4.4, um conjunto de grânulos  $\gamma_m^i$ , com  $i = 1, 2, \dots, l$ , é criado para cada uma das  $m$  variáveis selecionadas pelo algoritmo RecPun. Os grânulos  $\gamma_m^i$  são definidos por intervalos *fuzzy* com funções de pertinências  $A_{jm}^i \forall_{j,m}$ , com  $j = 1, 2, \dots, h$ . As



funções de pertinência usadas são do tipo triangular ou trapezoidal, construídas a partir da quádrupla  $\{l_{jm}^i, \lambda_{jm}^i, \Gamma_{jm}^i, L_{jm}^i\}$ , sendo  $l_{jm}^i$  e  $L_{jm}^i$  os limites inferiores e superiores dos intervalos fuzzy e  $\lambda_{jm}^i$  e  $\Gamma_{jm}^i$  valores intermediários para essas funções. Para funções trapezoidais  $\lambda_{jm}^i < \Gamma_{jm}^i$ , e para funções triangulares  $\lambda_{jm}^i = \Gamma_{jm}^i$ .

De acordo com Leite *et al.* (2009), um novo grânulo é criado quando um padrão de entrada  $x_{jm}^t \notin [l_{jm}^i, L_{jm}^i] \forall j, i$ , para algum  $m$ , ou quando  $x_{jm}^t \in [l_{jm}^i, L_{jm}^i] \forall j$  para algum  $i$ , mas o rótulo de classe  $c_k$  do padrão de entrada difere da classe associada a  $\gamma_m^i$ . Porém, no trabalho aqui proposto, somente a primeira condição é considerada, por tratar-se de um agrupamento não supervisionado, objetivado como etapa inicial do sistema. O novo grânulo  $\gamma_m^i$  é criado com base nas funções triangulares  $A_{jm}^i \forall j, m$ , com centro em  $x_m^t \in \mathfrak{R}^n$ . As funções de pertinências  $A_{jm}^i \forall j$  do novo grânulo assumem os parâmetros  $l_{jm}^i = x_{jm}^t - \frac{\rho_{jm}}{2}$ ,  $\lambda_{jm}^i = \Gamma_{jm}^i = x_{jm}^t$  e  $L_{jm}^i = x_{jm}^t + \frac{\rho_{jm}}{2}$ , conforme mostrado na Figura 4.5(a). Se o novo padrão de entrada  $x_{jm}^{t+\Delta}$ , para um inteiro positivo  $\Delta$ , estiver dentro dos limites atuais de algum  $\gamma_m^i$ , os parâmetros  $\lambda_{jm}^i$  e  $\Gamma_{jm}^i \forall j$  são atualizados para acomodar  $x_{jm}^{t+\Delta} \forall j$ . A atualização consiste em fazer  $\lambda_{jm}^i = x_{jm}^{t+\Delta}$  se  $x_{jm}^{t+\Delta} \in [l_{jm}^i, \lambda_{jm}^i]$  ou  $\Gamma_{jm}^i = x_{jm}^{t+\Delta}$  se  $x_{jm}^{t+\Delta} \in [\Gamma_{jm}^i, L_{jm}^i]$ , conforme mostrado na Figura 4.5(b).



**Figura 4.5:** Criação e Adaptação de Grânulos [adaptado de Leite *et al.* (2009)].

O parâmetro  $\rho_m \in X_m \in \mathfrak{R}^n$  define o tamanho máximo que um grânulo pode assumir no espaço de entrada  $X_m$  para o atributo  $m$ , com  $X_m$  normalizado para  $[0, 1]$ . O valor de  $\rho_m$  define a transparência e a precisão do modelo em estudo e uma boa estimativa para  $\rho_m$  é condicionada a limites aceitáveis para as classes definidas. De modo geral, quanto maior o valor de  $\rho_m$  menos grânulos são criados e menor é a precisão da não linearidade entre classes. Em situações opostas

valores menores de  $\mu$  podem provocar sobreposição e dificuldade de interpretação dos resultados (Leite *et al.*, 2009).

As funções de pertinência  $A_{jm}^i \forall j, m$ , com  $j = 1, 2, \dots, h$ , mostradas na Figura 4.5 e representadas pela função de distribuição de probabilidades  $\mu(x_{jm}^i)$  definida pela Equação 4.7:

$$\mu(x_{jm}^i) = \begin{cases} 0 & \text{se } x_{jm}^t \leq l_{jm}^i \text{ ou } x_{jm}^t \geq L_{jm}^i \\ 1 & \text{se } \lambda_{jm}^i \leq x_{jm}^t \leq \Gamma_{jm}^i \\ \frac{x_{jm}^t - l_{jm}^i}{\lambda_{jm}^i - l_{jm}^i} & \text{se } l_{jm}^i \leq x_{jm}^t \leq \lambda_{jm}^i \\ \frac{L_{jm}^i - x_{jm}^t}{L_{jm}^i - \Gamma_{jm}^i} & \text{se } \Gamma_{jm}^i \leq x_{jm}^t \leq L_{jm}^i \end{cases} \quad (4.7)$$

Com base no algoritmo descrito acima, a cada padrão de entrada  $x^t = [a_1, a_2, \dots, a_m]$  são obtidos  $m$  pontos focais  $\beta^i$  e  $m$  graus de compatibilidade  $\theta^i$ , postos como saídas dos grânulos  $\gamma_{a_m}^i$  que acomodam  $a_1, a_2, \dots, a_m$ , respectivamente. O ponto focal  $\beta_{a_m}^i \forall m$  corresponde ao padrão  $x_{jm}^i$ , para algum  $i$  e  $j$ , com menor desvio em relação ao padrão médio (centroide) dos padrões de entrada  $x_{jm}^i \forall j$ , ponderados pelos graus de pertinência obtidos pela função de distribuição de probabilidades,  $\mu(x_{jm}^i)$ . O ponto focal  $\beta_m^i \forall m$  é definido pelas Equações 4.8 e 4.9:

$$\beta_{a_m}^i = \operatorname{argmin} \left( \|x_{jm}^i - \bar{X}_m^i\| \right) \forall j \text{ e algum } i \text{ e } m \quad (4.8)$$

$$\bar{X}_m^i = \frac{\sum_{j=1}^h x_{jm}^i \mu(x_{jm}^i)}{\sum_{j=1}^h \mu(x_{jm}^i)} \quad (4.9)$$

O padrão médio  $\bar{X}_m^i$  compreende ao centro de massa do conjunto *fuzzy* resultante do processo de “desfuzzificação” do tipo centroide para os  $x_{jm}^i \forall j, m$  (Mendel, 2001).

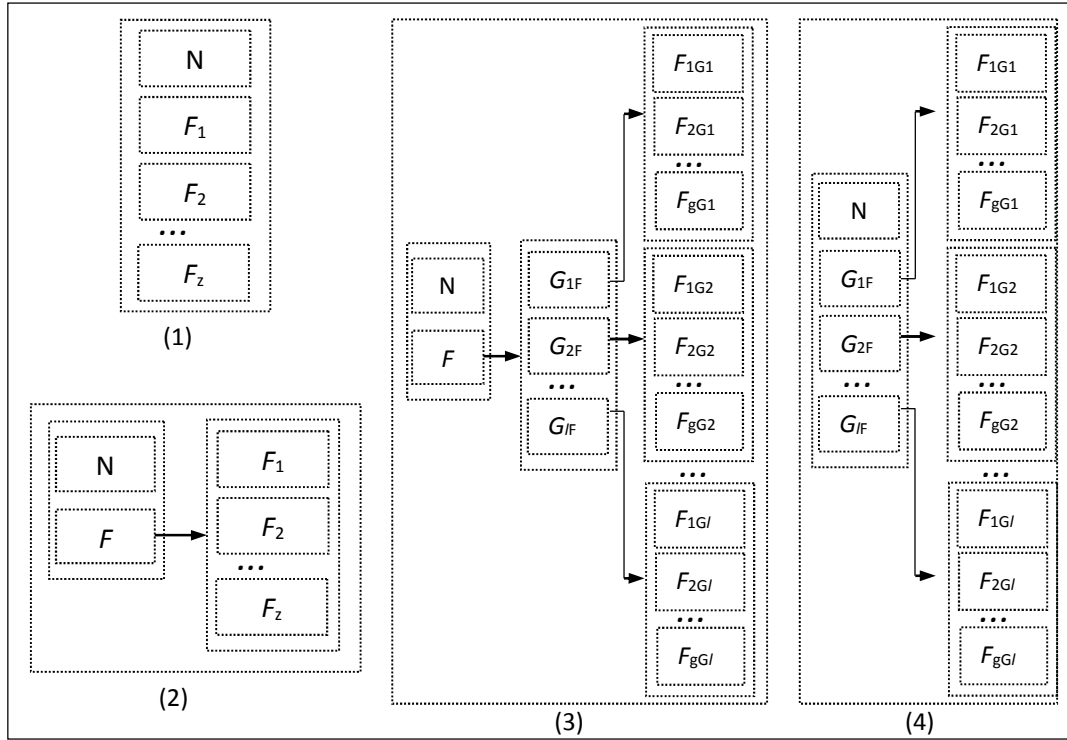
#### 4.3.4 Estrutura da Rede Neural sem Pesos

Conforme já mencionado, a RNSP pode realizar a detecção e o diagnóstico de falhas de forma paralela (Figura 4.6(1)) ou em cascata (Figuras 4.6(2) a 4.6(4)) (Pedrycz *et al.*, 2004), conforme arquiteturas mostradas na Figura 4.6. Essas redes são compostas por uma, duas ou três camadas de discriminadores (classificadores) com a estrutura da rede WiSARD. Os discriminadores são constituídos por MPLN (Myers e Aleksander, 1988, 1989). As arquiteturas mostradas nas Figuras 4.6(1) e 4.6(2) podem ser utilizadas em qualquer estudo de caso, já as mostradas nas Figuras 4.6(3) e 4.6(4) são específicas para casos em que as falhas estão classificadas em grupos (neste trabalho o estudo de caso TEP, apresentado na seção 4.4.1). Essas quatro arquiteturas são descritas como segue:

- ✓ Arquitetura 1 [Figura 4.6(1)]. A RNSP é composta por uma única camada contendo uma única WiSARD com  $k$  discriminadores treinados individualmente para representar uma única classe de padrões. Os estados de operação Normal ( $N$ ) e de Falhas ( $F_1, F_2, \dots, F_z$ ) são representados pelos discriminadores  $N, F_1, F_2, \dots, F_z$ , respectivamente. Na validação, o padrão testado é apresentado a todos os discriminadores e classificado como pertencente à

classe do discriminador com maior somatório de valores 1 como saídas das RAM, conforme modelo apresentado na Figura 4.3;

- ✓ Arquitetura 2 [Figura 4.6(2)]. A RNSP é composta por duas camadas. A primeira possui uma única WiSARD com dois discriminadores treinados individualmente para representar os padrões do estado de operação Normal ( $N$ ) e de Falha ( $F_1, F_2, \dots, F_z$ ). A segunda camada possui uma única WiSARD com  $z$  discriminadores treinados individualmente para representar os padrões de Falha ( $F_1, F_2, \dots, F_z$ ). Na validação, se a primeira camada classificar o padrão testado como de falha, o teste é realizado também na segunda camada, de modo a identificar o tipo de falha do padrão testado;
- ✓ Arquitetura 3 [Figura 4.6(3)]. A RNSP é composta por três camadas. A primeira possui uma única WiSARD com dois discriminadores, treinados individualmente para representar os padrões do estado de operação Normal ( $N$ ) e de Falha ( $F_1, F_2, \dots, F_z$ ). A segunda camada possui uma única WiSARD com  $l$  discriminadores, treinados individualmente para representar os  $l$  grupos de falhas ( $G_1, G_2, \dots, G_l$ ). A terceira camada possui  $l$  WiSARD com  $p$  discriminadores, treinados individualmente para representar os  $p$  estados de falhas ( $F_1, F_2, \dots, F_p$ ) considerados para cada grupo. Na validação, se a primeira camada classificar o padrão testado como de falha, o teste deverá ser realizado na segunda e terceira camadas, de modo a identificar o grupo e o tipo de falha, respectivamente. O teste na terceira camada é realizado apenas na WiSARD correspondente ao grupo de falhas apontado pela segunda camada;
- ✓ Arquitetura 4 [Figura 4.6(4)]. A RNSP é composta por duas camadas. A primeira possui uma única WiSARD com  $l + 1$  discriminadores, treinados individualmente para representar os padrões do estado de operação Normal ( $N$ ) e dos grupos de falhas ( $G_1, G_2, \dots, G_l$ ) considerados. A segunda camada possui  $l$  WiSARD com  $g$  discriminadores, treinados individualmente para representar os  $g$  estados de falhas ( $F_1, F_2, \dots, F_g$ ) considerados para cada grupo. Na validação, se a primeira camada classificar o padrão testado como pertencente a um grupo de falha  $l$ , o teste deverá ser realizado na segunda camada de modo a identificar o tipo de falha para o padrão de teste. O teste na segunda camada é realizado apenas na WiSARD correspondente ao grupo de falhas apontado pela primeira camada.



**Figura 4.6:** Arquiteturas da Rede Neural sem Pesos (RNSP).

Como os atributos são selecionados por classe, os discriminadores nas quatro arquiteturas da Figura 4.6 são estruturados para o aprendizado ou validação de padrões comportamentais, gerados com o processo de granulação *fuzzy* dos atributos selecionados. Conforme descrito no final da seção anterior, um padrão comportamental é a concatenação dos  $m$  pontos focais,  $\beta^t = \beta_{a_1}^i \beta_{a_2}^i \dots \beta_{a_m}^i$ , representados em binário simples e obtidos como saídas dos  $m$  grânulos vencedores. O padrão comportamental  $\beta^t$  corresponde à entrada da RNSP e, mais especificamente, às entradas para as RAM nos discriminadores. O tamanho de  $\beta^t$  em *bits* é  $m$  vezes o número de *bits* usados para representar  $\beta_{a_m}^t \forall m$ .

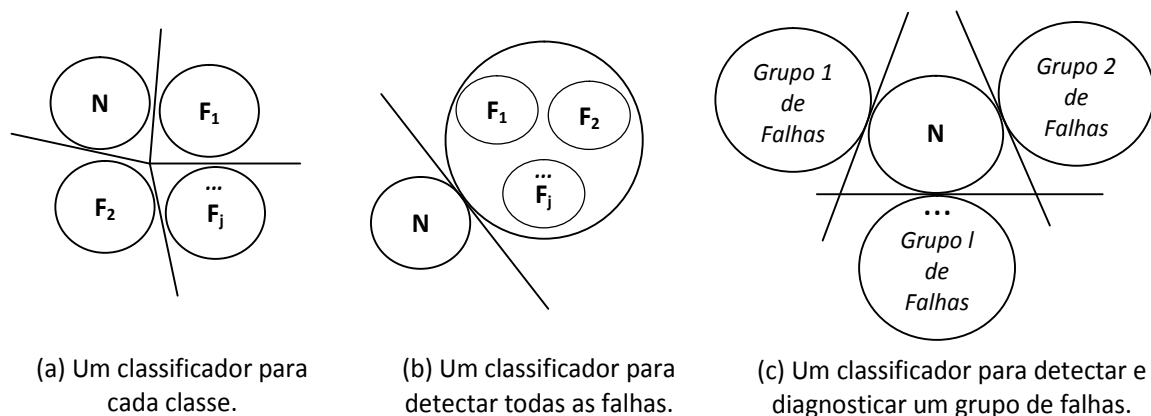
Conforme colocado anteriormente, os discriminadores são compostos por nodos RAM do tipo MPLN. Os conteúdos de memória em uma RNSP com nodos MPLN são incrementados ou decrementados, podendo chegar a 0 (limite inferior) ou a 1 (limite superior), conforme descrito na seção 4.2.1.4. Porém, na rede WiSARD os discriminadores são treinados apenas com padrões positivos das classes representadas. Combinando as duas ponderações, as RNSP com nodos MPLN consideradas neste trabalho são treinadas apenas com incrementos positivos nos conteúdos de memória de modo a produzir uma saída igual a 1 por um período de tempo determinado. No processo de treinamento, os conteúdos de memória acessados por  $\beta^t$  são atualizados com o valor definido pela função “*t*-norma produto” dos  $m$  graus de compatibilidade  $\theta_{a_m}^t$ , representada por  $\theta^t = \theta_{a_1}^i \wedge \theta_{a_2}^i \wedge \dots \wedge \theta_{a_m}^i$ , sendo  $\wedge$  o operador da função “*t*-norma produto” (Zadeh, 1994). No modelo proposto por Myers e Aleksander (1988) os conteúdos de memória na rede MPLN são inicializados por 0.5 (zero vírgula cinco), porém no trabalho aqui proposto esses conteúdos são inicializados por 0 (zero). Dessa forma, a cada padrão de entrada os conteúdos de memória acessados por  $\beta^t$  são  $C[a] = C[a] + \theta^t$ . De modo a tornar-se compatível com as saídas dos nodos e da rede WiSARD, a saída de um nodo MPLN aqui implementado é igual a 1 (um) se o conteúdo de memória  $C[a]$  for maior ou igual a um valor de *bleaching*

definido, caso contrário a saída é 0 (zero). Para isso o *bleaching* possui um valor real pertencente ao intervalo [0, 1], o que torna o nodo MPLN equivalente a um nodo RAM com *cut-point* (ponto de corte). Assim, um padrão testado é apresentado a todos os discriminadores e classificado como pertencente à classe do discriminador com maior somatório de valores 1 como saídas das RAM, como mostrado na Figura 4.3.

O nível de sensibilidade e especificidade do sistema para a detecção e o diagnóstico de falhas perpassa também por avaliações de outros parâmetros, tais como: i) variação no número de funções de pertinência no processo de granulação; ii) valor de  $\alpha$  (define a transparência e a precisão da não linearidade entre classes com a quantidade de grânulos criados); iii) representação binária dos pontos focais  $m$  definidos no processo de granulação para cada padrão processado (8, 12, 16 ou 32 *bits*); iv) tamanho dos padrões de entrada para a rede igual à multiplicação do item (iii) por  $m$  (número de atributos selecionados); v) número de entradas para as RAM igual a 2, 3, 4, 6 ou 8 *bits* [Figura 4.1(a)]; vi) número de RAM por discriminadores igual à divisão exata do item (iv) pelo item (v); vii) precisão na representação decimal dos conteúdos de memórias nas RAM; viii) uso do *bleaching* homogêneo ou heterogêneo; ix) dimensionalidade da aplicação; x) método de seleção de atributos (RecPun ou RecPun modificado); e xi) quantidade de atributos selecionados (1, 2, ...,  $m$ ). Os itens ix, x e xi não se aplicam a dados univariáveis.

#### 4.3.5 Filtro de Agrupamentos

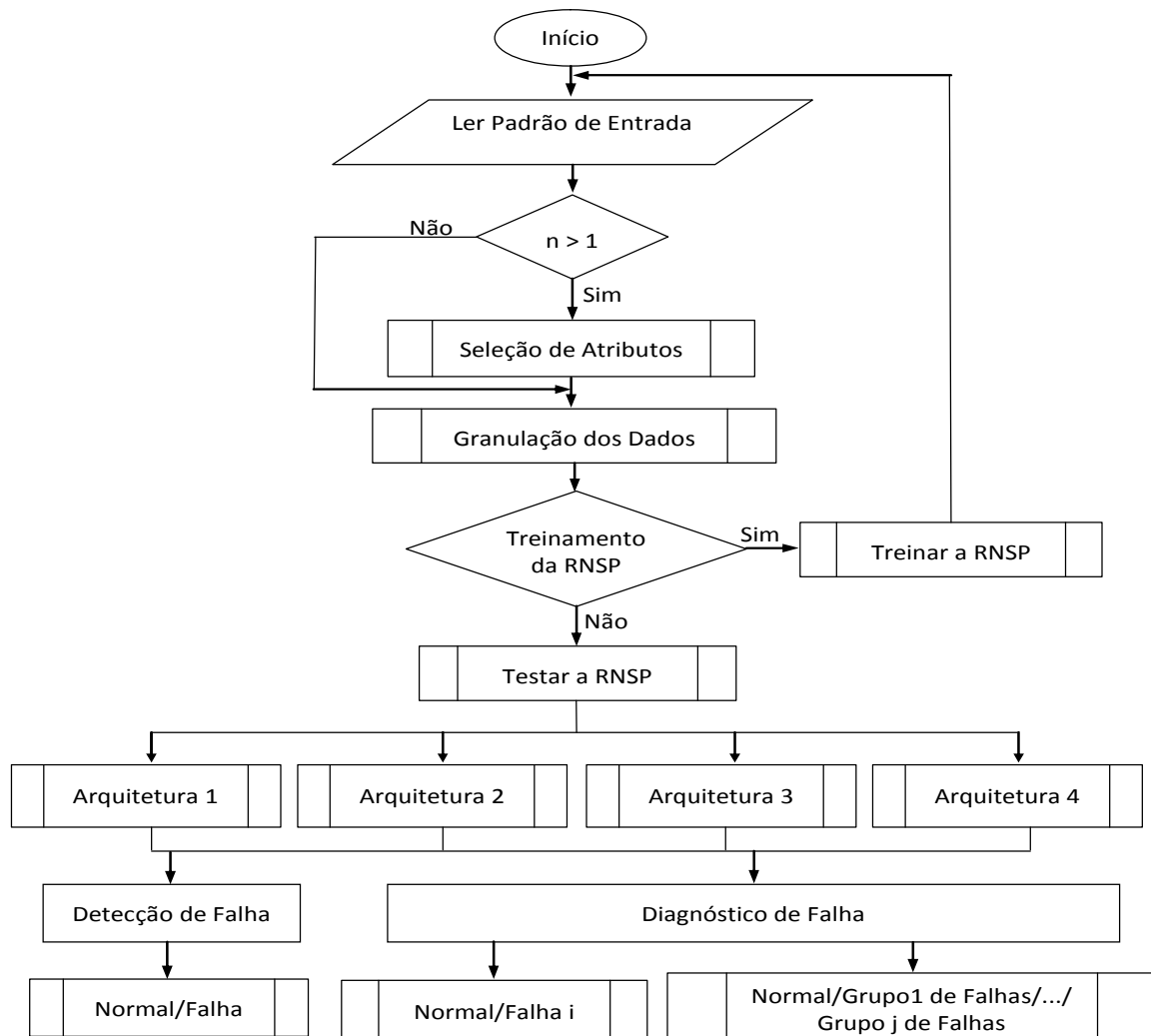
Conforme já mencionado, a rede WiSARD realiza a detecção e o diagnóstico de falhas simultaneamente e em único procedimento, com um diagnóstico por classe, conforme mostrado na Figura 4.7(a). Porém, opcionalmente e a partir dos resultados obtidos pela WiSARD, se uma classificação por grupo de falhas for requerida, a exemplo dos esquemas ilustrados nas Figuras 4.7(b) e 4.7(c), o filtro de agrupamento como etapa final do sistema FDD-RNSPG é usado. O esquema da Figura 4.7(b) classifica os padrões como estado de operação Normal ou estado de Falha, aqui referenciado como detecção de falhas, que na maioria dos sistemas de FDD é realizada em etapa anterior ao diagnóstico. Já o esquema da Figura 4.7(c) classifica os padrões de testes como de operação Normal ou pertencentes a um grupo de falhas com características comuns pré-definidas. Assim, é possível uma análise dos resultados em nível da rede (diagnóstico mais granulado) ou do sistema como um todo, a exemplo dos resultados e análises apresentados nas seções 4.4.1.1 e 4.4.2.1.



**Figura 4.7:** Esquemas de Detecção e Diagnóstico com o Sistema FDD-RNSPG.

### 4.3.6 Metodologia para o Treinamento e Teste do Sistema FDD-RNSPG

A Figura 4.8 sintetiza a metodologia utilizada para a detecção e o diagnóstico de falhas com o sistema FDD-RNSPG, onde  $n$  é o número de atributos para os padrões de entrada. A partir da definição de  $n$  para o primeiro padrão de um conjunto de treinamento ou validação, os demais apresentam a mesma dimensionalidade. Após as etapas de seleção e granulação dos atributos, a RNSP treinada ou testada apresenta uma das quatro arquiteturas mostradas na Figura 4.6.



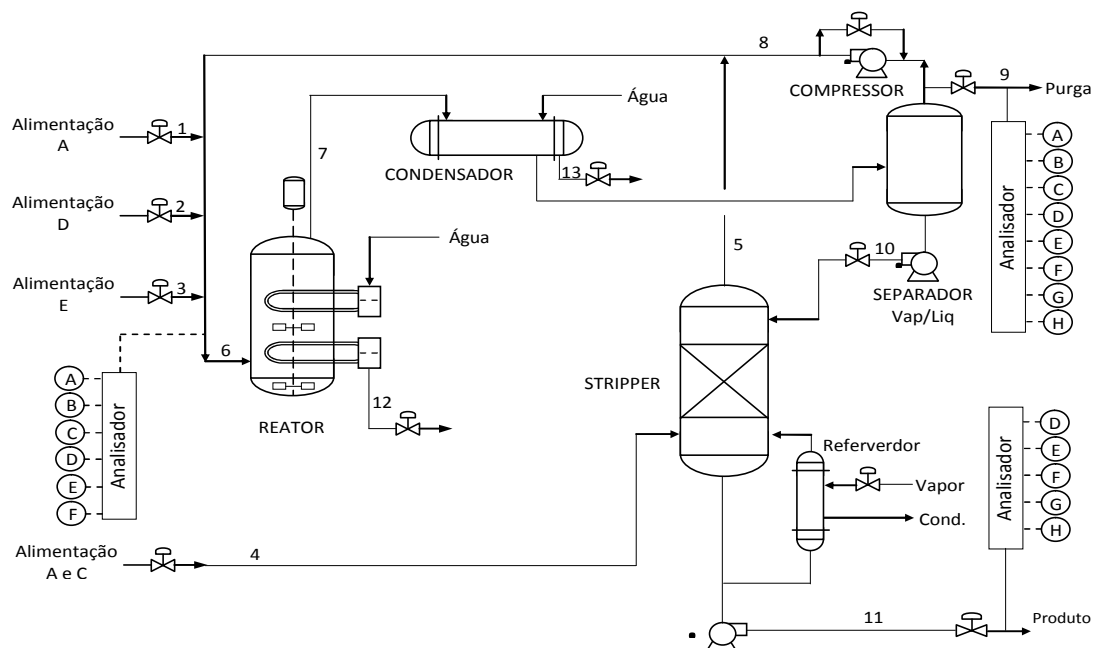
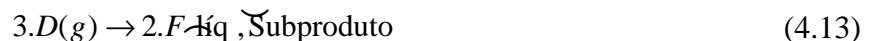
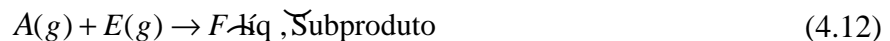
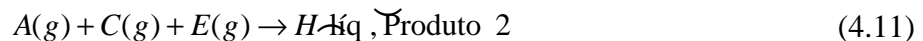
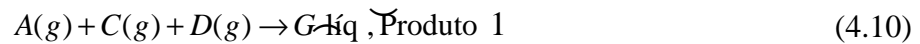
**Figura 4.8:** Metodologia para os Treinamentos e Testes com o Sistema FDD-RNSPG.

Nos testes realizados com os sistemas FDD desenvolvidos, os resultados são apresentados em termos de quantidade e taxa de acerto por classe/grupamento de padrões. A representação dos resultados apresentados pelos sistemas é feita por uma matriz de classificação com conteúdos variados. A escolha do sistema final para cada estudo de caso usou como critério a exatidão do sistema na detecção e no diagnóstico. Os percentuais de acertos/erros referem-se às classificações devidas/indevidas dos dados do conjunto de teste, com base nos limites estabelecidos para as classes consideradas. A exatidão dos sistemas, tanto para a detecção de falhas quanto para o diagnóstico, é obtida na saída do sistema como um todo (rede acoplada ao filtro de agrupamento). Conforme mostrado na Figura 4.8, o filtro aplicado na detecção de falhas classifica os padrões de teste como estado de operação normal ou estado de falha. Já o filtro usado para o diagnóstico das falhas considera os estados representados por operação normal ou estado de falha do tipo  $i$  ou grupo de falhas.

## 4.4 Estudos de Caso

### 4.4.1 Tennessee Eastman Process (TEP)

O *Tennessee Eastman Process* (TEP) foi proposto originalmente por Downs e Vogel (1993) como uma planta industrial em malha aberta com desafios para controle de processos. Ele também é considerado como um *benchmark* na área de detecção e diagnóstico de falhas, por caracterizar-se como um processo realista e com marcante não linearidade nos dados multivariáveis. O TEP compreende num arranjo reator/ separador/ reciclo centrado em cinco unidades principais de operação, como mostrado na Figura 4.9: reator; condensador de produto; separador líquido/vapor; compressor de reciclagem; e coluna de separação de produtos. O modelo TEP possui 12 variáveis manipuláveis e 41 variáveis medidas. Entre as variáveis medidas, 22 são amostradas a cada 3 minutos e correspondem às medições de vazão, temperatura e pressão. As demais variáveis medidas têm períodos de amostragem maiores, e consistem dos analisadores da alimentação para o reator, da purga e do produto final. O processo simulado no TEP objetiva a produção dos produtos *G* e *H* a partir dos reagentes *A*, *C*, *D* e *E*, gerando ainda um subproduto *F*, de acordo com as reações representadas abaixo. Além dos reagentes e produtos é utilizado também um inerte gasoso *B*, introduzido nas correntes de alimentação dos reagentes.



**Figura 4.9:** Diagrama Esquemático do *Tennessee Eastman Process* (TEP) (Oliveira *et al.*, 2017).

Como observado na Figura 4.9, o reator é alimentado por meio dos reagentes inseridos pelas correntes 1, 2 e 3. A mistura que sai do reator é conduzida para o condensador pela corrente 7, onde ocorre uma liquefação parcial por resfriamento. A saída do condensador é levada para a unidade de separação líquido/vapor onde uma fração da parte gasosa é purgada para evitar acumulação de inerte e subproduto no processo e a outra fração é enviada para o compressor

centrífugo e reciclada para o reator. A parte líquida segue para a *stripper* (coluna retificadora) pela corrente 10. Uma corrente (4) gasosa composta por A e C e por uma pequena quantidade do inerte B também é alimentada à *stripper* em contracorrente à alimentação líquida. Os reagentes removidos na *stripper* são reciclados ao reator e a parte líquida, constituída basicamente por G e H, consiste no produto final do processo.

Para este trabalho foi considerada a planta em malha fechada com o sistema de controle proposto por Ricker (1995). Esse autor propôs as condições ótimas de estado estacionário para os seis modos de operação do problema de desafio industrial proposto por Downs e Vogel (1993). O estado estacionário ótimo proposto por Ricker (1995), e relativo ao “caso base” de Downs e Vogel (1993) (razão mássica  $G:H$  de 50:50 e taxa de produção total igual a 14076 kg/h), foi usado na simulação dos dados para o treinamento e a validação do sistema FDD-RNSPG proposto. O trabalho original de Downs e Vogel (1993) considera 20 diferentes condições de falhas, mostradas na Tabela 4.1, e as 13 primeiras (em negrito) foram consideradas neste trabalho, pois as demais resultam de combinações de duas ou mais dessas 13 falhas ou de mudanças nos valores nominais das variáveis de entrada. As falhas do tipo oscilatório consistem em mudanças aleatórias e persistentes na variável em questão.

**Tabela 4.1:** Falhas Simuladas no TEP (Oliveira *et al.*, 2017)

Falha	Descrição	Alteração
<b>1</b>	<b>Razão A/C, mantendo B constante</b>	<b>Degrau</b>
<b>2</b>	<b>Composição de B, mantendo a razão A/C constante</b>	<b>Degrau</b>
<b>3</b>	<b>Temperatura de alimentação de D</b>	<b>Degrau</b>
<b>4</b>	<b>Temperatura da água de refrigeração para o reator</b>	<b>Degrau</b>
<b>5</b>	<b>Temperatura da água de refrigeração para o condensador</b>	<b>Degrau</b>
<b>6</b>	<b>Carga (válvula) de alimentação de A</b>	<b>Degrau</b>
<b>7</b>	<b>Carga (válvula) de alimentação de C</b>	<b>Oscilatória</b>
<b>8</b>	<b>Composições das alimentações de A, B e C</b>	<b>Oscilatória</b>
<b>9</b>	<b>Temperatura de alimentação de A</b>	<b>Oscilatória</b>
<b>10</b>	<b>Temperatura de alimentação de C</b>	<b>Oscilatória</b>
<b>11</b>	<b>Temperatura da água de refrigeração para o reator</b>	<b>Oscilatória</b>
<b>12</b>	<b>Temperatura da água de refrigeração para o condensador</b>	<b>Oscilatória</b>
<b>13</b>	<b>Variação na cinética da reação (velocidade, componentes e fatores)</b>	<b>Oscilatória</b>
14	Válvula de água de arrefecimento do reator	Desconhecida
15	Válvula de água de refrigeração do condensador	Desconhecida
16	Desconhecida	Desconhecida
17	Desconhecida	Desconhecida
18	Desconhecida	Desconhecida
19	Desconhecida	Desconhecida
20	Desconhecida	Desconhecida

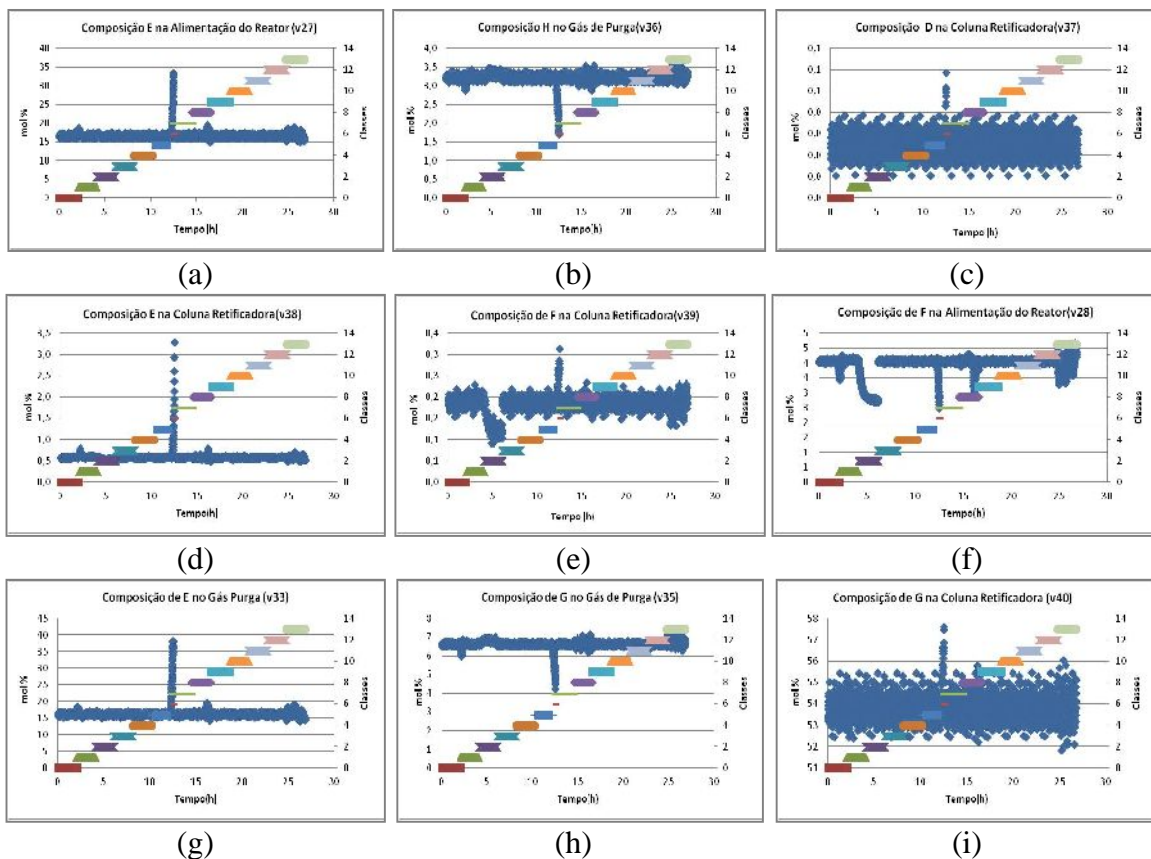
#### 4.4.1.1 Resultados e Análises - TEP

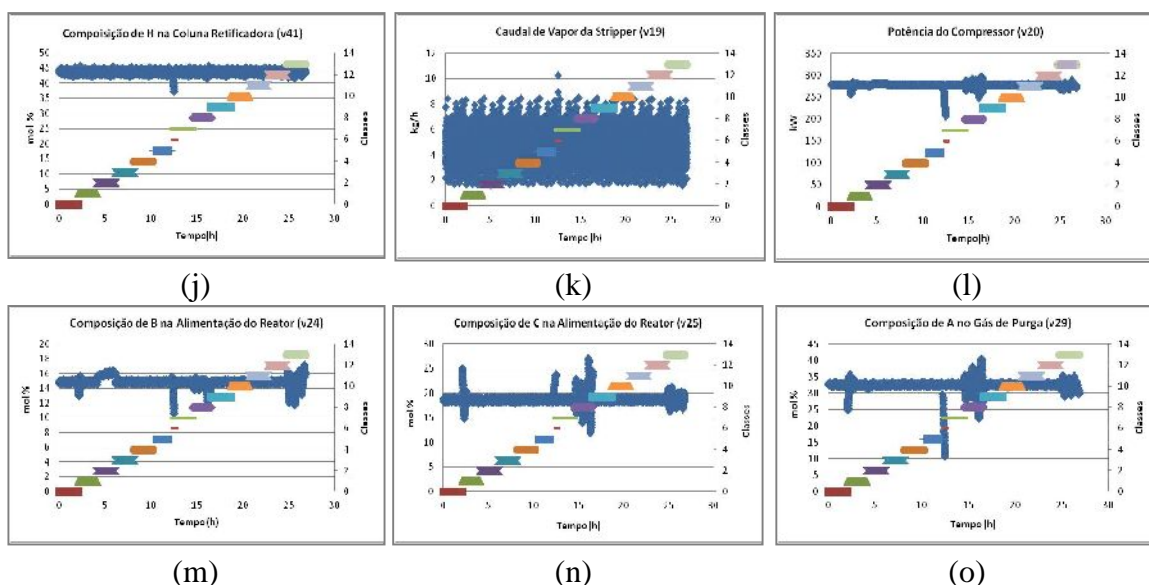
Para o treinamento e validação dos sistemas de FDD aqui desenvolvidos para o TEP foram gerados 5 (cinco) conjuntos de dados com o simulador proposto por Ricker (1995). Conforme mencionado anteriormente, a simulação dos dados teve como base o estado estacionário ótimo proposto por Ricker (1995), aqui denominado de “caso 1”, equivalente ao “caso base” proposto por Downs e Vogel (1993). Um desses conjuntos foi gerado com os valores originais e usado no treinamento das redes, e os outros 4 (quatro) foram gerados com variações de  $\pm 3\%$  e  $\pm 5\%$



na variável “taxa de produção” e usados na validação dos sistemas. Para Downs e Vogel (1993), a taxa de produção é a variável controlada de maior relevância na composição dos produtos finais *G* e *H* e admite variações no intervalo  $\pm 5\%$  de modo a considerar um mesmo estado estacionário da planta. Com exceção da falha 6, para cada configuração de dados foram realizadas 14 simulações de 1000 períodos de amostragem de 3 minutos, sendo uma simulação de operação normal e as outras referentes aos 13 tipos de falha considerados. Com o intuito de amenizar efeitos de inicialização foram descartados os 20 primeiros períodos de cada simulação (Oliveira *et al.*, 2017).

A título de ilustração, a Figura 4.10 mostra o conjunto de dados originais usados no treinamento para os modos de operação “Normal” e “Falha 4”, após a etapa de seleção de variáveis (atributos) com o algoritmo RecPun modificado e relativos à configuração de melhor desempenho com o sistema FDD-RNSPG. As “Figuras 4.10(a) a 4.10(e)” mostram variáveis comuns aos modos de operação Normal e Falha 4. As “Figuras 4.10(f) a 4.10(j)” são as demais variáveis relativas ao modo de operação Normal, e as “Figuras 4.10(k) a 4.10(o)” apresentam as outras variáveis selecionadas para o modo de operação Falha 4. No gráfico 4.10(a), por exemplo, a representação temporal da classe do padrão de treinamento, contendo a variável (atributo) selecionada (*v27*), está na mesma sequência temporal de entrada dessa e das demais variáveis. As cores representam as classes, de modo que uma mesma cor em todos os gráficos está relacionada a uma única classe. Para outros modos de operação os dados são semelhantes, mudando apenas as variáveis (atributos) selecionadas para cada um deles.





**Figura 4.10:** Dados Utilizados no Treinamento dos Sistemas FDD para o TEP: Modos de Operação “Normal” e “Falha 4”.

A Tabela 4.2 apresenta os percentuais de acerto no diagnóstico apresentado pelas RNSP granulares para a aplicação TEP. O percentual em cada célula da Tabela 4.2 corresponde ao melhor desempenho para uma dada configuração (modelo de arquitetura *versus* variação na variável taxa de produção), com os demais parâmetros descritos na seção 4.3.3. Na Tabela 4.3 é apresentado o conjunto de parâmetros para cada modelo de arquitetura mostrado na Tabela 4.2 com uma das taxas de produção de *G* e *H*, aquela considerada de melhor desempenho (em negrito).

**Tabela 4.2:** Percentuais (%) de Acerto das RNSP para a Aplicação TEP

Arquiteturas	Variação na Taxa de Produção				
	-5	-3	+3	+5	Média
1	99,11	<b>99,85</b>	99,32	97,75	98,86
2	98,90	<b>99,39</b>	99,20	97,61	98,78
3	95,35	<b>98,83</b>	92,03	84,05	92,57
4	95,45	<b>99,50</b>	92,16	84,20	92,83

**Tabela 4.3:** Melhores Parâmetros Encontrados pelo Sistema FDD-RNSPG com Base no Modelo de Arquitetura 1 com Taxa de Variação na Taxa de Produção para a Aplicação TEP

Arquiteturas	Parâmetros Avaliados								
	i	ii	iii	iv	v	vi	vii	viii	ix
1	-3	10	3	0,6	120	3	40	5	1
2	-3	10	3	0,6	120	3	40	5	2
3	-3	23	10	0,6	276	3	92	5	2
4	-3	23	10	0,6	276	3	92	5	2

i) variação na taxa de produção (%); ii) n° de variáveis selecionadas; iii) janela temporal = n° de funções de pertinência nos grânulos; iv) valor de (relação entre grânulos criados e precisão); v) padrão de entrada da rede (n° de *bits*); vi) entradas RAM (n° de *bits*); vii) n° de RAM por discriminador [(v)/(vi)]; viii) precisão dos conteúdos de memórias (n° de *casas decimais*); ix) *bleaching*: homogêneo (1) ou heterogêneo (2).

Com base nos dados das Tabelas 4.2 e 4.3, as seguintes considerações podem ser feitas:

- a) As quatro arquiteturas testadas apresentaram resultados satisfatórios para a aplicação TEP. Devido à proximidade dos seus modelos, as arquiteturas 1 e 2 apresentaram desempenhos semelhantes, comportamento também observado com as arquiteturas 3 e 4. A variação de -3% na taxa de produção apresentou maior consistência nas quatro arquiteturas, com um melhor desempenho para a arquitetura 1;
- b) Na arquitetura 1 a detecção e o diagnóstico de falhas são realizados simultaneamente. Nas arquiteturas 2, 3 e 4 a primeira camada realiza a detecção de falha enquanto as camadas 2 e 3, de acordo com a arquitetura, realiza o diagnóstico de falha;
- c) O parâmetro *bleaching* homogêneo apresentou melhor desempenho com a arquitetura 1, enquanto o *bleaching* heterogêneo se destacou nas arquiteturas 2, 3, e 4. De modo geral, o *bleaching* homogêneo é melhor quando as classes (modos de operação) consideradas apresentam quantidades iguais ou próximas de dados de treinamento;
- d) O desempenho dos sistemas de FDD é inversamente proporcional ao percentual de variação na taxa de produção, o que justifica a recomendação dada por Ricker (1995) em não usar variações fora do intervalo de  $\pm 5\%$ ;
- e) A precisão do nodo MPLN está na representação dos conteúdos de memórias com valores  $\in [0, 1]$ . Se o número de dígitos na representação decimal desses conteúdos for muito grande, a convergência implica em um período de tempo maior e em um aprendizado mais lento para um conjunto de treinamento com muitos exemplos. Para o TEP, a configuração de melhor desempenho foi obtida com uma precisão de 5 dígitos.

As matrizes de classificação apresentadas nas Tabelas 4.4 e 4.5 detalham o diagnóstico por classe da configuração WiSARD com nodos MPLN (com variação de -3% na taxa de produção), obtido na saída da rede com o uso dos algoritmos de seleção de atributos “RecPun modificado” e “RecPun original”, respectivamente (taxa de acertos na classificação de padrões em negrito).

**Tabela 4.4:** TEP: Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e Variação de -3% na Taxa de Produção e Uso do Algoritmo de Seleção de Atributos RecPun Modificado

Classes (Real)	Saídas Preditas pelo Sistema													Acertos (%)	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12		C13
C0	979								1						99,90
C1		980													100,00
C2			980												100,00
C3				980											100,00
C4			2		978										99,80
C5						980									100,00
C6		1					124		1						98,41
C7			5					969	6						98,88
C8			1						979						99,90
C9										978			2		99,80
C10											980				100,00
C11			1									979			99,90
C12													980		100,00
C13														980	100,00
<b>Percentual de Acerto da Rede</b>															<b>99,85</b>

**Tabela 4.5:** TEP: Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e Variação de -3% na Taxa de Produção e Uso do Algoritmo de Seleção de Atributos RecPun Original

Classes (Real)	Saídas Preditas pelo Sistema													Acertos (%)	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12		C13
C0	167			31	6	30		142		11	257	104	232		17,04
C1		979					1								99,90
C2			980												100,00
C3	169			31	8	31		138		11	258	97	237		3,16
C4	158			37	14	29		139		6	261	100	236		1,43
C5	160			31	9	31		138		11	263	98	239		3,16
C6		2					123		1						97,62
C7	161			24	11	27		198	2	24	228	103	202		20,20
C8			1						979						99,90
C9	162			44	20	43		144	1	19	267	88	192		1,94
C10	180			33	19	67		134		20	257	78	192		26,22
C11	179			25	19	31		132		14	266	100	214		10,20
C12	171			31	17	39		167		9	246	116	184		18,78
C13							1		1					978	99,80
<b>Percentual de Acerto da Rede</b>														<b>39,17</b>	

Comparando os diagnósticos nas Tabelas 4.4 e 4.5 percebe-se o quanto a modificação realizada no algoritmo RecPun foi importante para a detecção e o diagnóstico de falhas com o sistema FDD-RNSPG. A taxa de acertos na classificação de padrões com o algoritmo de seleção de atributos “RecPun original” e “RecPun modificado” é de 39,17% e 99,85%, respectivamente. A confusão na classificação se deve ao fato de o algoritmo RecPun original selecionar para a maioria das classes, na aplicação em estudo, os mesmos atributos. Para amenizar este fato, como já mencionado na seção 4.3.2, foi adicionado um passo ao final do algoritmo RecPun de modo que a relação biunívoca entre as classes permita no máximo 50% de atributos comuns.

De modo a demonstrar a boa adaptabilidade das RNSP para a detecção e diagnóstico de falhas com a planta TEP, os resultados aqui apresentados são comparados com os encontrados por (Oliveira *et al.*, 92017) e Soares e Galvão (2010). Oliveira *et al.* (2017) usaram os mesmos dados de treinamento e validação com um sistema de FDD baseado na WiSARD com nodos RAM padrão e nodos com contadores como conteúdos de memória. Enquanto Soares e Galvão (2010) abordaram a detecção e o diagnóstico de falhas no TEP com o uso de classificadores por grupos de falhas com características semelhantes. Para a construção dos classificadores Soares e Galvão (2010) utilizam métodos paramétricos (Análise Discriminante Linear - ADL e Análise Discriminante Quadrática - ADQ) e não paramétricos (k Vizinhos mais Próximos - kNN e Mínimos Quadrados Parciais - PLS). Nos resultados apresentados por Soares e Galvão (2010), a situação de melhor desempenho foi obtida com classificadores APS-ADQ para os 4 (quatro) grupos de falhas, conforme mostrado no final da Tabela 4.6. No entanto, os resultados obtidos pelo sistema FDD-RNSPG, aqui proposto, foram melhores do que os apresentados por Soares e Galvão (2010) e levemente superiores aos apresentados Oliveira *et al.* (2017). Para isso, a Tabela 4.6 apresenta também os principais indicadores de qualidade para o sistema FDD-RNSPG, considerando a rede WiSARD com nodos MPLN e variação de -3% na taxa de produção de *G* e *H* (melhores resultados em negrito).

**Tabela 4.6:** Indicadores de Qualidade na Validação do Sistema FDD-RNSPG com a Arquitetura 1 e Variação de -3% na Taxa de Produção de *G* e *H* e Comparação com Oliveira *et al.* (2017) e Soares e Galvão (2010)

Itens Observados (pRAM)	FDD-RNSPG		(Oliveira <i>et al.</i> 2017)		(Soares e Galvão, 2010)			
	RecPun Original	RecPun Modificado	RecPun Original	RecPun Modificado	Paramétricos		Não Paramétricos	
					APS-ADQ	APS-ADL	kNN	PLS
Padrões Testados	12866	12866	12856	12856	-	-		
Acertos - RNSP (n.)	5040	<b>12846</b>	5129	12788	-	-		
Acertos - RNSP (%)	39,17%	<b>99,85%</b>	39,90%	99,47%	-	-		
Filtro 1: Detecção de falhas (Média)	83,27%	99,99%	99,42%	<b>100,00%</b>	-	-		
• Estado de Operação Normal	17,04%	99,90%	1,02%	<b>100,00%</b>	-	-		
• Estado de Operação com Falha	88,73%	<b>100,00%</b>	99,42%	<b>100,00%</b>	-	-		
Filtro 2: Diagnóstico por Grupos de Falhas (Soares e Galvão, 2010)	47,98%	<b>99,90%</b>	58,02%	99,43%	93,20%	86,0%		
• Grupo 1 (falhas: 7, 8, 9 e 10)	57,98%	99,80%	78,44%	<b>99,87%</b>	94,1%	85,0%	71,7%	79,3%
• Grupo 2 (falhas: 1 e 4)	50,66%	<b>99,90%</b>	51,84%	98,32%	97,8%	90,7%	72,2%	77,3%
• Grupo 3 (falhas: 2, 3, 5, 6 e 11)	39,00%	<b>99,95%</b>	43,21%	99,85%	94,9%	84,8%	74,6%	80,8%
• Grupo 4 (falhas: 12 e 13)	59,29%	<b>100,00%</b>	53,83%	98,78%	86,0%	82,8%	70,6%	75,9%

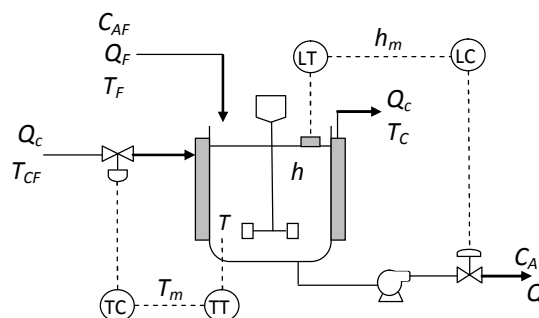
Além das comparações apresentadas na Tabela acima, os trabalhos apresentados por Lau, *et al.* (2013) e Barragan *et al.* (2016) também permitem uma comparação dos resultados aqui obtidos para o estudo de caso TEP. Lau, *et al.* (2013) propuseram um sistema de diagnóstico de falhas baseado em *Multi-Scale Principal Component Analysis* (MSPCA) e no *Adaptive Network-based Inference System* (ANFIS) e referenciado como MSPCA-ANFIS. Os percentuais de classificações corretas apresentados por Lau, *et al.* (2013) foram de 99,52% para os padrões de operação normal e 90,7% para os estados de falhas, excluindo as falhas 3, 9 e 15 das 21 falhas analisadas no processo TEP. No modelo aqui proposto esses percentuais foram de 99,90 e 100,00%, conforme mostrado na Tabela 4.6. Já o trabalho apresentado por Barragan *et al.* (2016) compreende num sistema de agrupamento de padrões com base no método *Fuzzy C-*

*Means* (FCM). Os dados utilizados foram gerados no simulador TEP, proposto por Ricker (1995). O estudo caso considera o estado de operação normal e o estado de falha 1 com base em 4 das 41 variáveis medidas no TEP. Os padrões de testes são agrupados em dois (Normal ou falha) ou três (Normal, falha ou híbridos) *clusters*. As métricas do SPCA<sub>ms</sub> tiveram uma taxa de classificação de sucesso de 95% para os *clusters* 1 (falha) e 2 (normal), e com SPCA tradicional, essas taxas foram de 70% e 80%, respectivamente.

Com os resultados e comparações acima, conclui-se que uma boa estrutura para o sistema FDD-RNSPG para o modelo TEP possui a seguinte configuração: “Seleção de Atributos” usando o algoritmo RecPun modificado com 10 (dez) atributos por classe; “Granulação dos Atributos Seleccionados” com 3 (três) funções de pertinência por grânulo; “Rede Neural sem Pesos” com uma única camada (arquitetura 1) e estrutura baseada na rede WiSARD com  $k$  discriminadores e nodos MPLN, padrões de entrada para a rede igual a 120 (10·12) *bits*, subpadrões de entradas para as RAM com 3 (três) *bits*, número de RAM por discriminador igual a 40 (120/3), uso do *bleaching* homogêneo, detecção do estado normal e de falha com diagnóstico por classe; “Filtro de Agrupamento” com diagnóstico por grupos de falhas, de acordo com os definidos por Soares e Galvão (2010) e mostrados na Tabela 4.6. O tempo de execução do treinamento e teste do sistema para a referida configuração foi de 13 (treze) segundos [os experimentos foram realizados numa única máquina básica da Intel (R), com processador core (TM) i5-2450M, CPU 2.50 GH, memória RAM de 4GB e sistema operacional de 64 *bits*]. Os dados de treinamento e validação foram simulados com base no modo de operação “caso 1” proposto por Ricker (1995) com variação de -3% na taxa de produção para o conjunto de validação.

#### 4.4.2 Reator de Tanque Agitado Continuamente (CSTR)

O reator tanque agitado Continuamente (CSTR, *Continuous Stirred Tank Reactor*) é não isotérmico, com resfriamento através de uma jaqueta e com nível de líquido variável. O CSTR é um problema multivariável e tem como base uma reação química clássica irreversível de primeira ordem do tipo  $A \rightarrow B$ . O modelo dinâmico do CSTR considera mistura perfeita e parâmetros físicos constantes (Singhal e Seborg, 2002). Um diagrama esquemático do CSTR e do seu sistema de controle é apresentado na Figura 4.11 (adaptado de Singhal e Seborg (2002)). Mais detalhes sobre a estrutura de controle e os parâmetros do controlador para o CSTR podem ser obtidos em Johannesmeyer (1999) e Singhal e Seborg (2002).



**Figura 4.11:** Diagrama Esquemático do Reator CSTR.

Na literatura relacionada, muitos estudos abordam a detecção e o diagnóstico de falhas para estudos de casos com reatores CSTR, apresentando um número significativo de possíveis condições de falhas (Sorsa e Koivo, 1992; Vaidyanathan e Venkatasubramanian, 1992; Singhal e Seborg, 2002). Como exemplo, o trabalho apresentado por Singhal e Seborg (2002) retrata uma base de dados típica para o estudo de caso CSTR, incluindo o estado de operação normal e 27 condições de falhas. Dentre as perturbações consideradas por Singhal e Seborg (2002),

destacam-se falhas e mudanças no processo, falhas de instrumentação e alterações nos ajustes de temperatura do reator.

O simulador aqui utilizado tem como referência o trabalho apresentado por Singhal e Seborg (2002), tendo sido simulados 18 dos 28 estados de operação considerados por esses autores. As 14 variáveis de processo e os 28 modos de operação considerados por Singhal e Seborg (2002) são apresentados, respectivamente, nas Tabelas 4.7 e 4.8. As 4 últimas variáveis/ medições da Tabela 4.7 são sinais de saída do controlador e não são mostradas na Figura 11. Os modos de operação aqui considerados estão em negrito na coluna 1 (um) da Tabela 4.8.

**Tabela 4.7:** Variáveis Medidas para o CSTR (Singhal e Seborg, 2002)

Variável	Descrição das Variáveis
$C_A$	Concentração da espécie A no reator (mol/L)
$T$	Temperatura do reator (K)
$T_c$	Temperatura do refrigerante na camisa de resfriamento (K)
$H$	Nível do líquido no reator (dm)
$Q$	Vazão de saída do reator (L/mim)
$Q_c$	Vazão do fluido refrigerante (L/mim)
$Q_F$	Vazão de alimentação do reator (L/mim)
$C_{AF}$	Concentração da espécie A na corrente de alimentação do reator (mol/L)
$T_F$	Temperatura de alimentação do reator (K)
$T_{CF}$	Temperatura de alimentação do fluido refrigerante (K)
$hC$	Medição do controlador de nível (mA); m = número de dados na janela atual
$QC$	Controlador de nível do reator (L/mim)
$TC$	Controlador de temperatura de saída (K)
$Q_cC$	Saída do controlador a partir do fluxo de resfriamento (mA)

**Tabela 4.8:** Modos de Operação para o CSTR (Singhal e Seborg, 2002)

Rótulos	Modos de Operação	Descrição	Valor Nominal
<b>N</b>	Operação normal	Operação nas condições normais, sem perturbações	N/A
<b>F1</b>	Desativação do catalisador	Rampa positiva na energia de ativação	A taxa da rampa para E/R é +3K/min
<b>F2</b>	Incrustação no trocador de calor	Rampa negativa na transferência de calor	A taxa da rampa para $U_{AC}$ é -125 J/(min·K)/min
<b>F3</b>	Medição do fluxo de líquido de resfriamento	A medição da vazão de resfriamento permanece com seu último valor	N/A
<b>F4 e F-4</b>	<i>Bias</i> na medição de temperatura do reator	A medição de temperatura do reator tem um <i>bias</i>	±4 K
<b>F5 e F-5</b>	<i>Stiction</i> (atrito estático) da válvula de resfriamento + F7	Banda morta para <i>stiction</i> = 5% do <i>span</i> da válvula	N/A

<b>F6 e F-6</b>	Mudança degrau em $Q_F$	Mudança degrau na vazão de alimentação	$\pm 10$ L/min
<b>F7 e F-7</b>	Mudança rampa em $C_{AF}$	Rampas na concentração de alimentação para cima ou para baixo	A taxa da rampa é $\pm 6 \cdot 10^4$ (mol/L)/min
<b>F8 e F-8</b>	Mudança rampa em $T_F$	Rampas na temperatura de alimentação para cima ou para baixo	A taxa da rampa é $\pm 0,1$ K/min
<b>F9 e F-9</b>	Mudança rampa em $T_{CF}$	Rampas na temperatura de alimentação de resfriamento para cima ou para baixo	A taxa da rampa é $\pm 0,1$ K/min
F10 e F-10	Mudança degrau em $P_{CU}$	Mudança degrau da pressão na linha de refrigeração	$\pm 2,5$ psi
F11 e F-11	Mudança degrau em $P_D$	Mudança degrau da pressão na linha de saída do reator	$\pm 5$ psi
<b>F12</b>	Oscilações amortecidas na vazão de alimentação	Mudanças na vazão de alimentação como $e^{-t/33} \cdot \text{sen}(2 \cdot \pi/10) \cdot \text{L/min}$	10 L/min
F13	Distúrbio auto-regressivo na vazão de alimentação	$Q_F(k) = 0,8 \cdot Q_F(k-1) + w(k); w(k) \sim N(0,1)^*$	N/A
<b>S1 e S-1</b>	Alteração do valor nominal de $T$	Alteração do valor nominal da temperatura do reator	$\pm 3$ K
<b>O1</b>	Oscilações de alta frequência na vazão de alimentação	Oscilações sustentadas com frequência de 3 ciclos/min	10 L/min
<b>O2</b>	Oscilações de frequência intermédia na vazão de alimentação	Oscilações sustentadas com frequência de 1 ciclo/min	10 L/min
<b>O3</b>	Oscilações de frequência intermédia na vazão de alimentação	Oscilações sustentadas com frequência de 0,5 ciclos/min	10 L/min
<b>O4</b>	Oscilações de baixa frequência na vazão de alimentação	Oscilações sustentadas com frequência de 0,2 ciclos/min	10 L/min

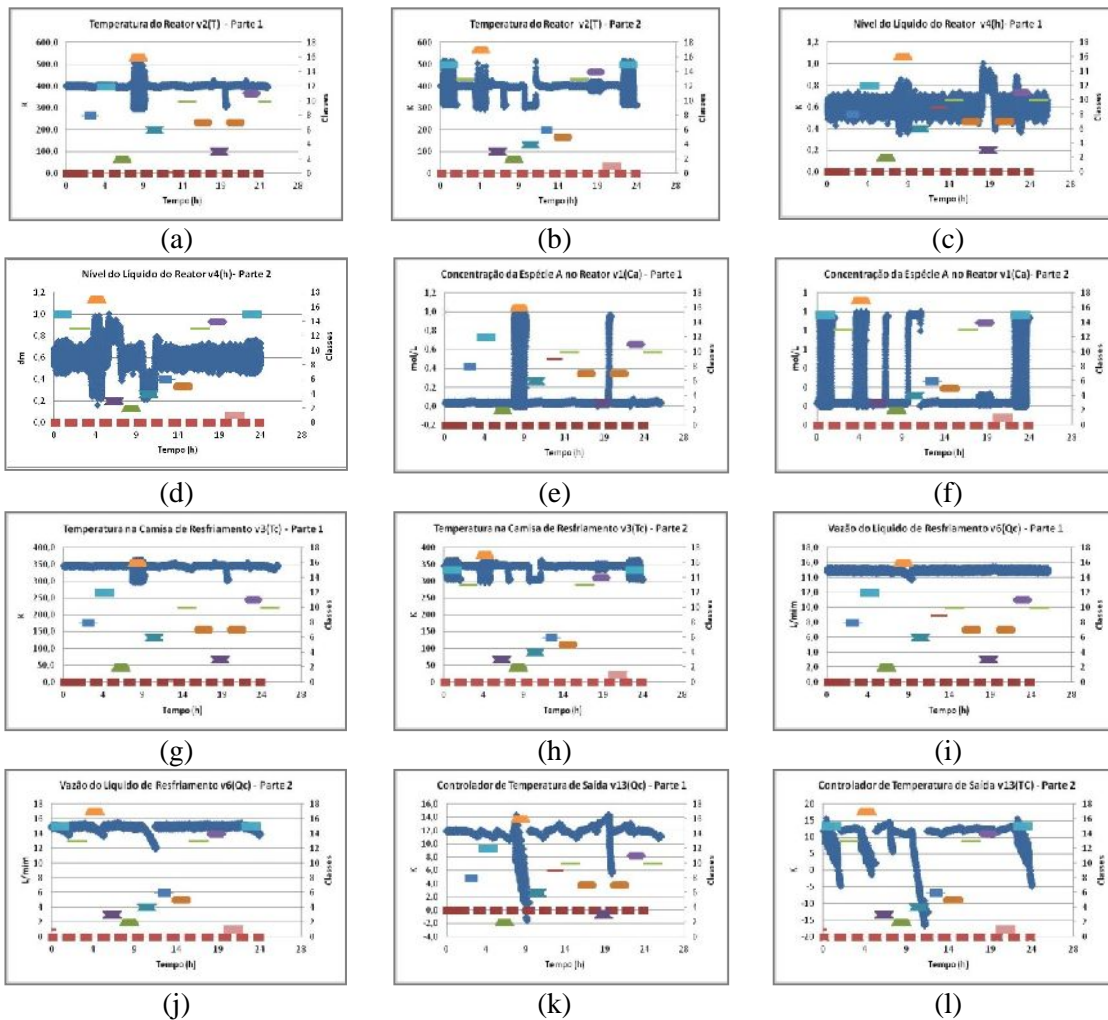
\* Curva normal com média 0 e desvio padrão 1.

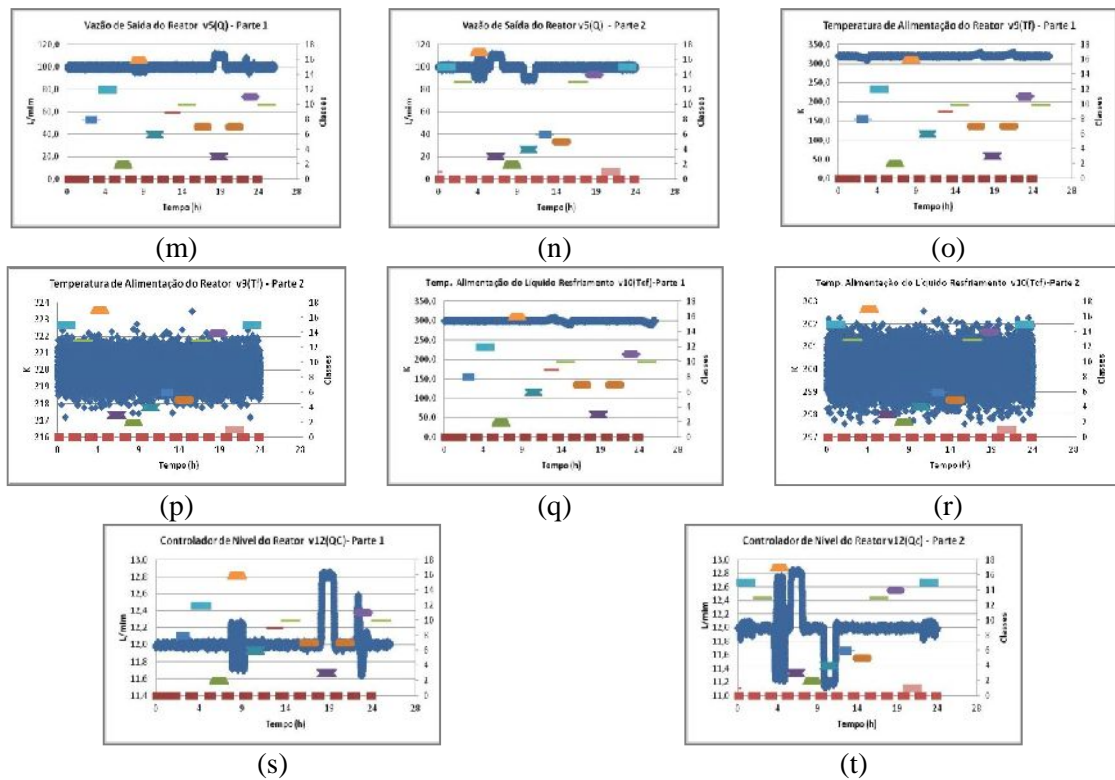
#### 4.4.2.1 Resultados e Análises - CSTR

Assim como em Singhal e Seborg (2002), para cada estado de operação (normal ou falha) foram simulados dados por um período de 120 minutos com amostragem de 5 segundos. Para os estados com falhas, a simulação nos 15 primeiros minutos acontece com base nos parâmetros nominais (*setpoints* para o estado normal). Nos 85 minutos seguintes o simulador é ajustado para o tipo de falha selecionado (não foram consideradas falhas simultâneas) e nos 20 minutos finais o simulador retoma os valores nominais de modo a estabilizar o reator para o estado normal. Assim, cada período de operação contém 1440 amostras com intervalos de 5 s. A composição final dos conjuntos de treinamento e de validação contém 1 (um) período de 120 minutos (1440 pontos) para o estado de operação normal e 1 (um) ou 2 (dois) períodos, também de 120 minutos, para cada estado de falha. Dos 1440 pontos referentes a um período de



simulação para um estado com falha, 420 (180 + 240, respectivamente relativos aos 15 e 20 minutos iniciais e finais para o estado de falha considerado) são rotulados como estado de operação normal e 1020 como o estado de falha considerado. A decisão por considerar um ou dois períodos para cada estado de falha e o ordenamento temporal dos diversos períodos de 120 minutos considerados foram definidos aleatoriamente. A título de ilustração, a Figura 4.12 mostra o conjunto de dados utilizados para o treinamento dos Sistemas FDD para o CSTR. Como na aplicação TEP, as Figuras 4.12(a) a 4.12(t) ilustram o conjunto de dados originais usados no treinamento para os modos de operação “Normal” e “Falha 4”, após a etapa de seleção de variáveis (atributos) com o algoritmo RecPun modificado. As variáveis apresentadas são relativas à configuração de melhor desempenho do sistema FDD-RNSPG para a aplicação CSTR. Como o volume de dados de treinamento é muito grande, cada variável foi representada em duas séries de dados, aqui denominadas de Parte 1 e Parte 2. Assim, as Figuras 4.12(a) a 4.12(d) são variáveis comuns aos modos de operação Normal e Falha 4, as Figuras 4.12(e) a 4.12(k) são as demais variáveis relativas ao modo de operação Normal e as Figuras 4.12(l) a 4.12(t) complementam as variáveis selecionadas para o modo de operação Falha 4.





**Figura 4.12:** Dados Utilizados no Treinamento dos Sistemas FDD para o CSTR: Modos de Operação “Normal” e “Falha 4”.

A partir da base de dados descrita acima, os resultados mostrados na Tabela 4.9 (em negrito os melhores resultados) referem-se aos indicadores de qualidade na detecção e diagnóstico de falhas para o sistema FDD-RNSPG, com base nas saídas da rede MPLN ou do sistema como um todo (rede mais filtro de agrupamento). Os valores ou percentuais apresentados para cada arquitetura correspondem ao melhor desempenho do sistema em relação aos parâmetros descritos na seção 4.3.3. Na Tabela 4.10 é apresentado o conjunto de parâmetros de melhor desempenho para cada modelo de arquitetura mostrado na Tabela 4.9. As arquiteturas 3 e 4 não foram aplicadas ao CSTR por não estar disponível na literatura um estudo de agrupamento de falhas para esse estudo de caso.

**Tabela 4.9:** Resultados das RNSP para a Aplicação CSTR

Parâmetros	Arquitetura 1	Arquitetura 2	Média
Padrões Testados (n°)	34560	34560	34560
Acertos (n°)	<b>34455</b>	34203	34329
Acertos (%)	<b>99,70</b>	98,97	99,34
<b>Deteção de Falhas (%)</b>	<b>99,95</b>	99,23	99,59
Estado Normal (%)	<b>99,96</b>	98,96	99,88
Estado de Falha (%)	<b>99,95</b>	99,36	99,82
<b>Diagnóstico de Falhas (%)</b>	<b>99,95</b>	99,36	99,66
F1 (%)	99,22	<b>99,31</b>	99,66
F2 (%)	99,41	<b>99,50</b>	99,26
F3 (%)	<b>100,00</b>	99,80	99,46
F4 (%)	99,85	<b>99,95</b>	99,90
F5 (%)	99,85	<b>99,91</b>	99,90
F6 (%)	<b>99,85</b>	99,80	99,88

F7 (%)	<b>99,75</b>	99,72	99,82
F8 (%)	<b>100,00</b>	99,80	99,74
F9 (%)	<b>100,00</b>	99,90	99,90
F10 (%)	<b>99,66</b>	99,55	99,95
F11 (%)	<b>99,31</b>	99,01	99,60
F12 (%)	<b>99,51</b>	86,86	99,16
F13 (%)	<b>98,92</b>	98,74	93,18
F14 (%)	99,02	<b>99,22</b>	98,83
F15 (%)	99,41	<b>98,71</b>	99,12
F16 (%)	99,46	<b>99,56</b>	99,06
F17 (%)	<b>100,00</b>	<b>100,00</b>	99,51
Tempo de Execução (s)	25	<b>18</b>	22

**Tabela 4.10:** Melhores Parâmetros Encontrados pelo Sistema FDD-RNSPG com Base na Arquitetura 1 na Aplicação CSTR

Arquiteturas	Parâmetros Avaliados							
	i	ii	iii	iv	v	vi	vii	viii
1	6	3	0,5	72	12	24	5	2
2	6	3	0,5	72	12	24	5	2

i) n° de variáveis selecionadas; ii) janela temporal = n° de funções de pertinência nos grânulos; iii) valor de (relação entre grânulos criados e precisão); iv) padrão de entrada da rede (n° de bits); v) entradas RAM (n° de bits); vi) n° de RAM por discriminador [(iv)/(v)]; vii) precisão nos conteúdos de memórias (n° de casas decimais); viii) *bleaching*: homogêneo (1) ou heterogêneo (2).

Com base nos dados das Tabelas 4.9 e 4.10 as seguintes considerações podem ser feitas:

a) As duas arquiteturas apresentaram resultados satisfatórios. Porém, a arquitetura 1 (um) com os parâmetros mostrados na Tabela 4.10 foi a configuração com o melhor desempenho na deteção e diagnóstico de falhas, com uma taxa de acerto na classificação de 99,70%;

b) A arquitetura 1 realiza a detecção e o diagnóstico de falhas de forma paralela e por classe. Para isso, cada discriminador é treinado para reconhecer os padrões de uma única classe. Assim, um padrão de entrada  $x$  em um tempo  $t$  é classificado como estado de operação normal ou estado de falha do tipo  $i$ , com  $i = 1, 2, \dots, j$ ;

c) Na arquitetura 2 a detecção e o diagnóstico de falhas são realizados em cascata. Desse modo, a primeira camada realiza a detecção de falha, classificando um padrão de entrada  $x$ , num tempo  $t$ , como estado de operação normal ou estado de falha. Se o padrão de entrada for considerado como estado de falha, a segunda camada realiza o diagnóstico de falha, classificando o referido padrão como falha do tipo  $i$ , com  $i = 1, 2, \dots, j$ ;

d) O parâmetro *bleaching* homogêneo apresenta desempenho inferior em relação ao *bleaching* heterogêneo nas duas arquiteturas testadas. Esse comportamento é devido à diferenciação no quantitativo de padrões de treinamento para o estado de operação normal e os demais estados relacionados aos tipos de falha considerados.

A matriz de classificação apresentada na Tabela 4.11 detalha o diagnóstico por classe obtido na saída da rede WiSARD com nodo MPLN e uso do algoritmo de seleção de atributos “RecPun modificado”. De modo comparativo, a matriz de classificação apresentada na Tabela 12 detalha o mesmo diagnóstico, porém com o uso do algoritmo de seleção de atributos “RecPun original” (acertos na classificação de padrões em negrito nas Tabelas 11 e 12).

**Tabela 4.11:** Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e o Algoritmo de Seleção de Atributos RecPun Modificado

Classe (Real)	Saídas Preditas com a RNSP-MPLN																		
	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	Acertos (%)
F0	11100																		100,00
F1		1016													4				99,61
F2			1020																100,00
F3				1020															100,00
F4					2040														100,00
F5						2019						21							98,97
F6							2040												100,00
F7								2040											100,00
F8									1020										100,00
F9										1020									100,00
F10											2040								100,00
F11						1						1019							99,90
F12													1020						100,00
F13														1020					100,00
F14															1020				100,00
F15																1020			100,00
F16																	2040		100,00
F17																		1020	100,00
<b>Percentual de Acerto da Rede</b>																			<b>99,88</b>

**Tabela 4.12:** Matriz de Classificação para o Sistema FDD-RNSPG com a Arquitetura 1 e o Algoritmo de Seleção de Atributos RecPun Original

Classe (Real)	Saídas Preditas com a RNSP-MPLN																		
	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	Acertos (%)
F0	11026	1	2	1	8	1	4		4	1			1		2	10	20	19	99,33
F1	1	1016												2			1		99,61
F2	4		638										378						62,55
F3				1018		1					1								99,80
F4					2036										1	1	1	1	99,80
F5		2				2032									6				99,61
F6		1					2037			1								1	99,85
F7	2							1804		218		6	8					2	88,43
F8	1	1							990		28								97,06
F9	1					1		3		1010		4	1						99,02
F10		4							177		1856	2						1	90,98
F11	1							3		17		998	1						97,84
F12	1							1					1018						99,80
F13		528												491		1			48,14
F14						1					1				1017	1			99,71
F15																1012	6	2	99,22
F16	4															2	2030	4	99,51
F17	11									1							1	1007	98,73
<b>Percentual de Acerto da Rede</b>																			<b>95,59</b>

Com os diagnósticos apresentados nas Tabelas 4.11 e 4.12, percebe-se o melhor desempenho do sistema FDD-RNSPG com o uso do algoritmo RecPun modificado. Conforme já mencionado, o algoritmo de seleção de atributos RecPun original seleciona conjuntos equivalentes ou semelhantes de atributos para duas ou mais classes, dentre as consideradas. Essa semelhança ou equivalência entre os atributos selecionados justifica a inferioridade nos resultados apresentados. Como já mencionado na seção 4.3.2 e no estudo de caso 1 (TEP), a modificação realizada no algoritmo RecPun permite no máximo 50% de atributos comuns entre duas das classes consideradas.

Como já mencionado, 17 das 27 condições operacionais de falhas consideradas por Singhal e Seborg (2002) foram consideradas no simulador CSTR desenvolvido e utilizado neste trabalho. O método proposto por Singhal e Seborg (2002) objetiva a correspondência de padrões em uma base de dados de séries temporais multivariada. O modelo proposto localiza períodos de dados semelhantes quando comparados a um período atual arbitrariamente selecionado. Para isso, utiliza-se um fator de similaridade PCA, associado a uma métrica proposta pelos autores para determinar de forma não supervisionada a distância entre dois conjuntos de dados. Para a validação da metodologia proposta, os dados são processados em fluxos sem a necessidade de treinamento do sistema. Diferentes métodos de correspondência de padrões são comparados em termos da precisão (número de registros corretamente identificados) e da eficiência (capacidade de localização de registros semelhantes na base de dados históricos) da pesquisa. As percentagens de precisão e eficiência são valores médios para os 28 modos operacionais considerados no estudo de caso do CSTR. Para a metodologia proposta, a precisão e eficiência são de 69% e 79%, respectivamente. Em média (74%), o desempenho do método proposto foi superior comparado com outros métodos estatísticos multivariados testados pelos autores. Embora com propósitos diferentes, a precisão na detecção e diagnóstico de falhas com o sistema FDD-RNSPG foi maior que a precisão apresentada por Singhal e Seborg (2002).

Com os resultados e as considerações feitas acima, uma estrutura proposta para o sistema FDD-RNSPG para o modelo CSTR possui a seguinte configuração: “Seleção de Atributos” por classe com o algoritmo RecPun modificado; “Granulação dos Atributos Selecionados” com o algoritmo de granulação *fuzzy* com 3 (três) funções de pertinência por grânulo; “Rede Neural sem Pesos” com a arquitetura 1 (um) e modelo WiSARD com nodos MPLN, padrões de entrada para a rede igual a 72 *bits* (12·6), subpadrões de entradas para as RAM com 6 (seis) *bits*, número de RAM por discriminadores igual a 12 (72/6), uso do *bleaching* homogêneo, diagnóstico por classe com as saídas da rede; “Filtro de Agrupamento” como etapa final do sistema foi utilizado apenas para a classificação de um padrão de entrada como estado de operação normal ou estado de falha, pois nenhum agrupamento de falhas foi considerado para o modelo CSTR. O tempo de execução do treinamento e teste do sistema para essa configuração foi de 22 segundos.

## 4.5 Conclusões Parciais

Este trabalho abordou o problema de detecção e diagnóstico de falhas em sistemas dinâmicos com o uso de Redes Neurais Sem Pesos (RNSP). O modelo proposto foi testado com os estudos de casos TEP e CSTR, ambos multivariáveis e com dados simulados. Com os resultados obtidos conclui-se que as RNSP adaptaram-se bem para o problema de detecção e diagnóstico de falhas em sistemas dinâmicos multivariável. A boa adaptação só foi possível mediante a contribuição: da seleção de atributos (variáveis) por classe; do processo de granulação *fuzzy* dos atributos selecionados; do armazenamento de probabilidades nos conteúdos de memórias nos nodos MPLN; do uso da técnica de *bleaching* para amenizar o uso da aleatoriedade entre os

discriminadores da rede WiSARD; e dos filtros de agrupamentos na definição de diagnósticos por grupos de falhas.

O uso do algoritmo RecPun (RECompensa/ PUNição) com a modificação proposta foi de extrema importância para a definição dos limites entre as classes consideradas. A *diferenciação* nos atributos entre classes, associada às habilidades *fuzzy* no processo de granulação dos atributos selecionados permitiram a geração de padrões comportamentais parecidos (próximos nas suas representações binárias) para dados de uma mesma classe. Essa proximidade nas representações binárias fortalece o processo de treinamento dos discriminadores e, conseqüentemente, o reconhecimento de padrões próximos por um mesmo discriminador na rede WiSARD.

O potencial das RNSP, mais especificamente da rede WiSARD, no processo de reconhecimento e classificação de padrões está diretamente relacionado com: o tipo de armazenamento dos conteúdos de memórias nas RAM; na fidelidade à metodologia de apresentação dos padrões de entradas para a rede e, mais especificamente, para os discriminadores; e na técnica de *bleaching* usada para amenizar o uso da aleatoriedade no desempate entre os discriminadores da rede.

De acordo com a estrutura e funcionalidade da rede WiSARD, cada discriminador reconhece e classifica padrões como se pertencessem a uma única classe. Para isso, é treinado um discriminador com padrões considerados como estado de operação normal e um discriminador para cada tipo/grupo de falha(s) considerado. Desse modo, é possível afirmar que a detecção e o diagnóstico de falhas são realizados simultaneamente. Esse procedimento simultâneo difere da maioria dos modelos propostos para o problema de FDD, onde uma estrutura é usada para a detecção de falhas e, posteriormente, outra estrutura é usada para o diagnóstico das falhas detectadas. Também, o treinamento e teste na rede WiSARD são realizados em um único procedimento (única apresentação), conferido assim uma consistência nos resultados obtidos. Ainda, a rede WiSARD pode ser facilmente adaptada para lidar com novos tipos de falhas.

Em geral, o modelo proposto foi projetado para o problema FDD em sistemas dinâmicos multivariados. Os estudos de caso serviram apenas para avaliar a qualidade do sistema FDD-RNSPG. No entanto, a configuração do sistema está relacionada às especificidades de cada aplicação, tais como: dimensionalidade, estados operacionais considerados (classes ou grupos), tipos de dados, formatação das bases de dados, entre outros. Além disso, como a rede WiSARD é composta de discriminadores, treinados individualmente para cada classe ou grupo considerado, a inclusão de novos tipos de falhas ou falhas múltiplas é apenas uma questão de adaptação do sistema. Em particular, os resultados mostraram o sucesso dos sistemas desenvolvidos para a detecção e diagnóstico de falhas, com taxas de sucesso de 99,85% e 99,70% para as aplicações TEP e CSTR, respectivamente.

## Abreviaturas

ADL:	Análise Discriminante Linear.
ADQ:	Análise Discriminante Quadrática.
AG:	Algoritmo Genético.
ANFIS:	Adaptive Network-based Inference System.
APS:	Algoritmo de Projeções Sucessivas.
CAPES:	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.
CNPq:	Conselho Nacional de Desenvolvimento Científico e Tecnológico.



CPU:	<i>Central Processing Unit.</i>
CSTR:	<i>Continuous Stirred Tank Reactor.</i>
<i>Disc<sub>k</sub></i> :	Discriminador <i>k</i> .
FDD:	<i>Fault Detection and Diagnosis.</i>
FDD- RNSPG:	Sistema de Detecção e Diagnóstico de Falhas com Rede Neural Sem Peso Granular.
kNN:	<i>k-Nearest Neighbour.</i>
MPLN:	<i>Multiple-valued Probabilistic Logic Neuron.</i>
MSPCA:	<i>Multi-Scale Principal Component Analysis.</i>
PLN:	<i>Probabilistic Logic Neuron.</i>
<i>Pun<sub>i</sub></i> :	Punição para o atributo <i>i</i> .
QA:	Quantidade de Atributos (variáveis) considerados em um estudo de caso.
QC:	Número de classes consideradas em um estudo de caso.
RAM:	<i>Random Access Memories.</i>
<i>Rec<sub>i</sub></i> :	Recompensa para o atributo <i>i</i> .
RecPun:	RECompensa/ PUNição.
RNA:	Rede Neural Artificial.
RNSP:	Rede Neural Sem Pesos.
RP:	Recompensa/Punição.
SPCA <sub>ms</sub> :	Fator de Similaridade PCA <i>Multiscale</i> .
TEP:	<i>Tennessee Eastman Process.</i>
WiSARD:	<i>Wilkes-Stonham-Aleksander Recognition Device.</i>

## Notações

$2^p$ :	Quantidade de memória na RAM locadas por $p$ bits de entradas.
$a_n^t$ :	Atributo $n$ do padrão de entrada $x^t$ .
$A_{jm}^i$ :	Função de pertinência $j$ para o atributo $m$ no grânulo $i$ .
$l_{jm}^i$ :	Limite inferior para o atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$L_{jm}^i$ :	Limite superior para o atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$\aleph^k$ :	Conjunto dos números naturais com $k$ classes.
$\Re^n$ :	Conjunto dos números reais.
$x^t$ :	Padrão de entrada $x$ no instante $t$ .
$x_{jm}^t$ :	Atributo $m$ para o padrão de entrada $x$ no tempo $t$ e função de pertinência $j$ .

$x_{jm}^{t+\Delta}$ : Atributo  $m$  para o padrão de entrada  $x$  no tempo  $t+$  e função de pertinência  $j$ .  
 $A(g)$ : Reagente gasoso utilizado nos simuladores TEP e CSTR.  
 $a$ : Endereço formado pelo padrão binário  $I$ .  
 $B(g)$ : Inerte gasoso utilizado no simulador TEP e produto no simulador CSTR.  
 $C$ : Reagente utilizado no simulador TEP.  
 $C[a]$ : Conteúdo de memória acessado pelo vetor de entrada  $a$ .  
 $C_A$ : Concentração da espécie  $A$  no reator.  
 $C_{AF}$ : Concentração da espécie  $A$  na corrente de alimentação do reator.  
 $C_k$ : Classe relacionada ao discriminador  $k$ .  
 $D(g)$ : Reagente utilizado no simulador TEP.  
 $d$ : Diferença entre a primeira e segunda maiores pontuações por um discriminador.  
 $E(g)$ : Reagente utilizado no simulador TEP.  
 $F(g)$ : Subproduto gerado no simulador TEP.  
 $f(x)$ : Função que define aproximadamente os limites para as classes consideradas.  
 $F_j$ : Padrão de Falha do tipo  $j$  para a aplicação utilizada.  
 $g(r)$ : Valores desejados como saídas para o nodo MPLN.  
 $G$ : Produto de saída no simulador TEP.  
 $G_l$ : Grupo  $l$  de falhas.  
 $h$ : Nível de líquido no reator.  
 $H$ : Produto de saída no simulador TEP.  
 $hC$ : Medição do controlador de nível.  
 $i$ : Valores possíveis de uma variável.  
 $I$ : Vetor de entradas para as RAM.  
 $K$ : Medida de temperatura em Kelvin.  
 $k$ : Quantidade de discriminadores na rede WiSARD.  
 $m$ : Número de atributos selecionados pelo algoritmo RecPun.  
 $n$ : Dimensionalidade dos vetores de entrada para os estudos de caso considerados.  
 $N$ : Estado de operação normal para os estudos de caso considerados.  
 $p$ : Número de *bits* do vetor de entradas para as RAM.  
 $P_{2,max}$ : Segunda maior pontuação obtida por um dos discriminadores da rede  
: WiSARD.  
 $P_{max}$ : Pontuação máxima obtida por um dos discriminadores da rede WiSARD.  
 $Q$ : Vazão de saída do reator.  
 $Q_c$ : Vazão do fluido refrigerante.

$QC$ :	Controlador de nível do reator.
$Q_cC$ :	Saída do controlador a partir do fluxo de resfriamento.
$Q_F$ :	Vazão de alimentação do reator.
$r$ :	Número de RAM em um discriminador.
$R_{i,c}$ :	Posição do atributo $i$ na classe $c$ .
$RP_i$ :	Recompensa e Punição para o atributo $i$ .
$T$ :	Temperatura do reator.
$T_c$ :	Temperatura do refrigerante na camisa de resfriamento.
$TC$ :	Controlador de temperatura de saída.
$T_{CF}$ :	Temperatura de alimentação do fluido refrigerante.
$T_F$ :	Temperatura de alimentação do reator.
$u$ :	Valor indefinido no nodo PLN.
$V_{i,c}$ :	Valor que representa a importância do atributo $i$ para a classe $c$ .
$x$ :	Padrão de entrada para o sistema.
$X_m$ :	Conjunto de variáveis $m$ para o padrão de entradas $x$ .
$y$ :	Saída da rede neural sem pesos com valor igual a um rótulo de classe.
$z$ :	Quantidade de falhas consideradas no estudo de caso.

## Símbolos Gregos

$\beta^t$ :	Padrão comportamental do padrão de entrada $x$ no tempo $t$ .
$\beta_{a_m}^i$ :	Ponto focal do atributo $m$ no grânulo $i$ .
$\beta^i$ :	Ponto focal para o grânulo $i$ .
$\Delta$ :	Variação com valor inteiro positivo.
$\gamma_m^i$ :	Grânulo $i$ para o atributo $m$ .
$\Gamma_{jm}^i$ :	Valor intermediário do atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$\lambda_{jm}^i$ :	Valor intermediário do atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$\Lambda$ :	Operador da função t-norma produto.
$\mu(x_{jm}^i)$ :	Grau de pertinência do atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$\theta_{a_m}^i$ :	Grau de compatibilidade do atributo $m$ no grânulo $i$ .
$\theta^i$ :	Grau de compatibilidade do grânulo $i$ .
$\theta^t$ :	Grau de compatibilidade do padrão de entrada $x$ no tempo $t$ .
:	Taxa de aprendizado para o nodo MPLN.
:	Parâmetro que define a precisão dos grânulos criados.
:	Operador de somatório de valores definidos.

## Referências

- Aleksander, I. (1967). Adaptive systems of logic networks and binary memories. *Proceedings of the Spring Joint Computer Conference, ACM, 30*, 707-712.
- Aleksander, I., Morton, H. (1990). An introduction to neural computing. *Chapman and Hall*, London.
- Aleksander, I., Thomas, W. V., Bowden, P. A. (1984). WISARD: A radical step forward in image recognition. *Sensor Review, 4(3)*, 120-124.
- Angelov, P., Yager, R. (2011). A new type of simplified fuzzy rule-based system, *International Journal of General Systems, 41(2)*, 163-185.
- Angelov, P., Zhou, X. (2006). Evolving fuzzy systems from data streams in real-time. *International Symposium on Evolving Fuzzy Systems*, 29-35.
- Araújo, L. A. (2011). RWISARD: um modelo de rede neural sem peso para reconhecimento e classificação de imagens em escala de cinza. *Dissertação de Mestrado*, Programa de Pós-Graduação em Engenharia de Sistemas e Computação. Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.
- Bandeira, L. C., França, H. L., França, F. M. G. (2009). NC-WiSARD: Uma interpretação Booleana da arquitetura neocognitron. *In: Anais do IX Congresso Brasileiro de Redes Neurais e Inteligência Computacional*, CD-Rom, Ouro Preto, Minas Gerais, Brasil, 25-28.
- Barragan, J. F., Fontes, C. H., Embiruçu, M. (2016). A wavelet-based clustering of multivariate time series using a Multiscale SPCA approach. *Computers & Industrial Engineering, 95*, 144-155.
- Blázquez, L. F., Miguel, L. J. (2005). Additive fault detection in nonlinear dynamic systems with saturation. *ISA Transactions, 44(4)*, 515-538.
- Bledsoe, W. W., Browning, I. (1959). Pattern recognition and reading by machine. *Proceedings Eastern Joint Computer Conference, 225-232*.
- Burns, C. M. (2006). Towards proactive monitoring in the petrochemical industry. *Safety Science, 44(1)*, 27-36.
- Cardoso, D. D. O., Lima, P. M., de Gregório, M., Gama, J., França, F. M. (2011). Clustering data streams with weightless neural networks. *European Symposium on Artificial Neural Networks, ESANN2011, Computational Intelligence and Machine Learning*, Bruges, Belgium.
- Cardoso, D. O., Carvalho, D. S., Alves, D. S., Souza, D. F., Carneiro, H. C., Pedreira, C. E., França, F. M. (2016). Financial credit analysis via a clustering weightless neural classifier. *Neurocomputing, 183*, 70-78.
- Cardoso, D., De Gregorio, M., Lima, P., Gama, J., & França, F. (2012). A weightless neural network-based approach for stream data clustering. *International Conference on Intelligent Data Engineering and Automated Learning*, Springer Berlin Heidelberg, 328-335.
- Carneiro, H. C., França, F. M., e Lima, P. M. (2015). Multilingual part-of-speech tagging with weightless neural networks. *Neural Networks, 66*, 11-21.
- Carvalho, D. S., Carneiro, H. C., França, F. M., Lima, P. M. (2013). B-bleaching: agile overtraining avoidance in the WiSARD weightless neural classifier. *European Symposium on Artificial Neural Networks, ESANN2013, Computational Intelligence and Machine Learning*, 515-520.

- Chen, T. Y. (2013). An interactive method for multiple criteria group decision analysis based on interval type-2 fuzzy sets and its application to medical decision making. *Fuzzy Optimization and Decision Making*, 12(3), 323-356.
- Chiang, L. H., Russel, E. L., Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. 1. Edição, Springer.
- Conti, V., Militello, C., Vitabile, S., Sorbello, F. (2009). An embedded fingerprints classification system based on weightless neural networks. *Frontiers in Artificial Intelligence and Application. Proceedings of the 18th Conference on New Directions in Neural Networks: Workshop Italian on Neural Networks, WINN2009, IOS Press Editor*, 193, 67-75.
- De Gregorio, M., Giordano, M. (2014). Change detection with weightless neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 403-407.
- De Souza, A. F., Freitas, F. D., De Almeida, A. G. C. (2010). High performance prediction of stock returns with VG-RAM weightless neural networks. *Workshop on High Performance Computational Finance, WHPCF2010, IEEE*, 1-8.
- Downs, J. J., Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3), 245-255.
- Fan, C. F., Yih, S., Tseng, W. H., Chen, W. C. (2013). Empirical analysis of software-induced failure events in the nuclear industry. *Safety Science*, 57, 118-128.
- Grieco, B. P., Lima, P. M., De Gregorio, M., França, F. M. (2010). Producing pattern examples from mental images. *Neurocomputing*, 73(7), 1057-1064.
- Guilford, J. P. (1950). *Fundamental statistics in psychology and education*. 4<sup>a</sup> Edição, New York: McGraw-hill Book.
- Hong, T., Wang, P. (2014). Fuzzy interaction regression for short term load forecasting. *Fuzzy Optimization and Decision Making*, 13(1), 91-103.
- Johannesmeyer, M. C (1999). Abnormal situation analysis using pattern recognition techniques and historical data. *M.Sc. Thesis, University of California, Santa Barbara, CA*.
- Kan, W. K., Aleksander, I. (1987). A probabilistic logic neuron network for associative learning. *In Proc. of the IEEE International Conference on Neural Networks 2*, San Diego, California, 541-548.
- Kuncheva, L. I., Whitaker, C. J. (2003). Examining the relationship between majority vote accuracy and diversity in bagging and boosting. *Technical Report School of Informatics, University of Wales, Bangor*.
- Lau, C. K., Ghosh, K., Hussain, M. A., Hassan, C. C. (2013). Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS. *Chemometrics and Intelligent Laboratory Systems*, 120, 1-14.
- Leite, D. P., Costa Jr., F. Gomide (2009). Evolving granular classification neural networks. *The 2012 International Joint Conference on Neural Networks, IEEE*, 1, 1736-1743.
- Leite, D. P., Costa, P., Gomide, F. (2012). Evolving granular neural network for fuzzy time series forecasting. *The 2012 International Joint Conference on Neural Networks, IEEE*, 1-8.

- Li, D. F. (2010). A new methodology for fuzzy multi-attribute group decision making with multi-granularity and non-homogeneous information. *Fuzzy Optimization and Decision Making*, 9(1), 83-103.
- Liu, H. T. (2007). An improved fuzzy time series forecasting method using trapezoidal fuzzy numbers. *Fuzzy Optimization and Decision Making*, 6(1), 63-80.
- Ludermir, T. B., Carvalho, A. C. P. L. F., Braga, A. P., de Souto, M. C. P. (1999). Weightless neural models: a review of current and past works. *Neural Computing Surveys*, 2, 41-61.
- McCulloch, W. S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-137.
- McElroy, B., Howells, G. (2011). Automated adaptation of input and output data for a weightless artificial neural network. *International Journal of Database Theory and Application*, 4(3), 49-58.
- Mendel, J. M. (2007). Advances in Type-2 Fuzzy Sets and Systems. *Information Sciences*, 177(1), 84-110.
- Mpofu, N. (2006). Forecasting stock prices using a weightless neural network. *Journal of Sustainable Development in Africa*, 8(1), 115-119.
- Myers, C., Aleksander, I. (1988). Learning algorithms for probabilistic logic nodes. *In Abstracts of I Annual INNS Meeting*, Boston, 205-205.
- Myers, C., Aleksander, I. (1989). Output functions for probabilistic logic nodes. *In Proceedings of the IEE International Conference Neural Networks*, London, 310-314.
- Nurmaini, S., Hashim, S. Z. M., Jawawi, D. N. A. (2009). Modular weightless neural network architecture for intelligent navigation. *International Journal of Advanced in Soft Computing and its Application*, 1(1), 1-18.
- Pedrycz, W., Gomide, F. (2007). Fuzzy systems engineering: Toward human-centric computing. *Wiley-IEEE Press, 1ª Edição*, 526p.
- Pedrycz, W., Reformat, M., Han, C. W. (2004). Cascade architectures of fuzzy neural networks. *Fuzzy Optimization and Decision Making*, 3(1), 5-37.
- Pedrycz, W., Vukovich, W. (2001). Granular neural networks. *Neurocomputing*, 36, 205-224.
- Pradhan, S. K., Parhi, D. R., Panda, A. K. (2006). Neuro-fuzzy technique for navigation of multiple mobile robots. *Fuzzy Optimization and Decision Making*, 5(3), 255-288.
- Ricker, N. L. (1995). Optimal steady-state operation of the Tennessee Eastman challenge process. *Computers and Chemical Engineering*, 19(9), 949-959.
- Romano, D., Kinnaert, M. (2006). Robust design of fault detection and isolation systems. *Quality and Reliability Engineering International*, 22(5), 527-538.
- Singhal, A., Seborg, D. E. (2002). Pattern matching in multivariate time series databases using a moving-window approach. *Industrial & Engineering Chemistry Research*, 41(16), 3822-3838.
- Sirlantzis, K., Howells, G., Gherman, B. (2009). Novel modular weightless neural architectures for biometrics-based recognition. *Proceedings of the 17th European Symposium on Artificial Neural Networks, ESANN'2009, Advances in Computational Intelligence and Learning*, Bruges, Belgium.

- Soares, A. S., Galvão, R. K. H. (2010). Fault detection using linear discriminant analysis with selection of process variables and time lags. *International Conference on Industrial Technology, ICIT2010*, IEEE, 217-222.
- Sorsa, T.; Koivo, H.(1992). Application of artificial neural networks in process fault diagnosis. *Automatica*, (29), 843-849.
- Subhashini, R., Nagarajan, E. (2014). Detection of an incognitos intruder in industries and semantic mapping of emotions. *International Journal of Applied Engineering Research*, 9(20), 6727-6734.
- Tsymbal, A., Cunningham, P., Pechenizkiy, M., Puuronen, S. (2003). Search strategies for ensemble feature selection in medical diagnostics. *Proceedings of the 16th Symposium on Computer-Based Medical Systems, CBMS2003*, IEEE, 124-129.
- Vaidyanathan, R., Venkatasubramanian, V. (1992). Representing and diagnosing dynamic process data using neural networks. *Engineering Applications of Artificial Intelligence*, (5), 11-21.
- Vale, K. M. O., Neto, A. F., Canuto, A. M. P. (2010). Using a reinforcement-based feature selection method in classifier ensemble. *Proceedings of the 10th International Conference on Hybrid Intelligent Systems, HIS2010*, IEEE, 213-218.
- Yang, S. K., Liu, T. S. (1998). A Petri net approach to early failure detection and isolation for preventive maintenance. *Quality and Reliability Engineering International*, 14(5), 319-330.
- Zadeh, L. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3), 77-84.

## CHAPTER V

### Chapter 5: Faults Detection and Diagnosis with Probabilistic Weightless Neural Networks

#### Abstract

This chapter approaches the problem of Faults Detection and Diagnosis in multivariate dynamic systems with the use of probabilistic weightless neural networks. Initially, the variables go through a selection process that determines the variables of greater relevance for each class considered in the application. The selected attributes are granulated by a recursive fuzzy inference system based on the concepts of clouds and on the group structures. For each selected attribute a set of clouds is generated and, for each input pattern, a cloud is chosen the selected attribute. The selected clouds form behavioral patterns, which are presented to the weightless neural networks used. The faults detection and diagnosis by class or groups of failures are obtained directly at the outputs of the tested networks or by a grouping filter, as the final step of the system. The probabilistic weightless neural systems were validated by two case studies with simulated data. One of them corresponds to a simulation of an industrial plant, known in the literature as “Tennessee Eastman Process (TEP)”. The other simulates a Continuous Stirred Tank Reactor (CSTR). The results show the success of the system for the detection and diagnosis of failures with percentage of precision in the classification of up to 99.89% and 99.88% for the respective applications.

**Keywords:** Weightless Neural Networks; Faults Detection and Diagnosis; Fuzzy Inference System; Clouds and Attribute Selection.



## CAPÍTULO V

### Capítulo 5: Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Probabilísticas

#### Resumo

Este capítulo aborda o problema de detecção e diagnóstico de falhas em sistemas dinâmicos multivariáveis com uso de redes neurais sem pesos probabilísticas. Inicialmente as variáveis passam por um processo de seleção que determina as variáveis de maior relevância para cada classe considerada na aplicação. Os atributos selecionados são granulados por um sistema de inferência *fuzzy* recursivo baseado nos conceitos de *clouds* e nas estruturas de grupo. Para cada atributo selecionado é gerado um conjunto de *clouds* e a cada padrão de entrada uma *cloud* é escolhida para o atributo selecionado. As *clouds* selecionadas formam padrões comportamentais, os quais são apresentados à rede neural sem pesos probabilística. A detecção e o diagnóstico de falhas por classe ou grupos de falhas são obtidos diretamente nas saídas das redes testadas ou por um filtro de agrupamento, como etapa final do sistema. O sistema neural sem pesos probabilístico foi validado por dois estudos de casos com dados simulados. Um deles corresponde a uma simulação de uma planta industrial, conhecida na literatura como *Tennessee Eastman Process* (TEP). O outro simula um reator tanque agitado continuamente (CSTR, *Continuous Stirred Tank Reactor*). Os resultados mostram o sucesso do sistema para a detecção e o diagnóstico de falhas com percentuais de acertos na classificação de 99,89% e 99,88% para as respectivas aplicações.

**Palavras-chave:** Redes Neurais sem Pesos; Detecção e Diagnóstico de Falhas; Sistema de Inferência *Fuzzy*; *Clouds* e Seleção de Atributos.

## 5.1 Introdução

O desenvolvimento de sistemas de detecção e diagnóstico de falhas (FDD, *Fault Detection and Diagnosis*) é uma área de grande interesse da comunidade científica. Dispor de mecanismos para a detecção antecipada e segura de falhas em sistemas dinâmicos, testados em modelos representativos de processos reais, assegura uma maior confiabilidade nos equipamentos, evita perdas materiais e, até mesmo, acidentes de trabalho e ambientais (Blázquez e Miguel, 2005; Burns, 2006; Chiang *et al.*, 2001; Fan *et al.*, 2013; Romano e Kinnaert, 2006; Yang e Liu, 1998).

Para Chiang *et al.* (2001), a oportunidade em desenvolver sistemas de FDD aumenta cada vez mais com o aumento da instrumentação dos processos industriais, principalmente quando um grande volume de dados é disponível para monitoramento, sob a forma de históricos do processo. Porém, como o armazenamento de dados é um processo oneroso, é crescente o desenvolvimento de sistemas com aprendizado incremental a partir de séries de dados históricos (Hong e Wang, 2014; Liu, 2007), com destaque para os sistemas neuro-*fuzzy* evolutivos (Angelov e Zhou, 2006; Fernandez *et al.*, 2015; Kar *et al.*, 2014; Leite *et al.*, 2009; Leite *et al.*, 2012; Pedrycz e Gomide, 2007; Silva *et al.*, 2014). Nos sistemas evolutivos, além dos ajustes nos parâmetros característicos dos sistemas adaptativos, também acontecem adaptações nas suas estruturas com a evolução do processamento dos dados (Angelov e Zhou, 2006; Fernandez *et al.*, 2015; Leite *et al.*, 2009; Silva *et al.*, 2014). Sistemas neuro-*fuzzy* combinam as habilidades de aprendizagem e aproximação de funções complexas pelas redes neurais com o bom desempenho dos sistemas *fuzzy* para o processamento de dados imprecisos (Lee e Lee, 1975). Neste contexto, este trabalho aborda a detecção e o diagnóstico de falhas em processos industriais dinâmicos baseados na “seleção de atributos”, “sistemas de inferência *fuzzy*” e “redes neurais sem pesos”. O sistema FDD aqui proposto é validado por dados multivariáveis simulados, relacionados a dois estudos de casos.

A seleção de atributos (variáveis) usada neste trabalho é uma adaptação do algoritmo RecPun (RECompensa/ PUNição) proposto por Vale *et al.* (2010). O referido algoritmo utiliza-se de recompensa e punição para ordenar os atributos com base na sua importância para cada classe considerada na aplicação. A proposta do modelo é ter, no mínimo, um classificador por classe e que os atributos selecionados para uma dada classe sejam importantes somente para esta classe. Já o sistema de inferência *fuzzy* usado realiza a granulação dos atributos selecionados com base no conceito de *clouds* (“nuvens”) proposto em Angelov e Yager (2011) e no algoritmo usado por Rosa *et al.* (2013), com adaptações para dados multivariáveis. As *clouds* criam uma partição dos dados de entrada de modo “recursivo” e com base nas “densidades” locais e globais para o processamento do fluxo de dados relativo aos atributos selecionados. Por último, as “redes neurais” usadas são consideradas granulares, pois aprendem padrões comportamentais definidos pelo sistema de inferência *fuzzy* na etapa anterior. Essas redes têm as suas estruturas definidas com base no número de classes consideradas na aplicação e na quantidade de atributos selecionados. No sistema proposto, a evolução acontece apenas com a etapa de “granulação *fuzzy*” dos atributos selecionados.

As redes neurais usadas pertencem a uma classe conhecida como “Redes Neurais Sem Pesos” (RNSP), propostas inicialmente por Aleksander (1967). Essas redes são modelos digitais baseados em dispositivos de memória de acesso randômico ou aleatório (RAM, *Random Access Memories*). O aprendizado nas RNSP acontece em “memórias” inseridas no próprio neurônio, em forma de tabelas-verdade. Inicialmente, os modelos neurais sem pesos eram totalmente booleanos e voltados quase que exclusivamente para o reconhecimento e a classificação de imagens digitais (Aleksander, 1967). Porém, esses modelos evoluíram para valores reais, tanto nas entradas e saídas das RAM quanto no armazenamento dos conteúdos de memórias, a

exemplos das RAM probabilísticas (pRAM, *probabilistic RAM*) (Gorse e Taylor, 1988). Os nodos pRAM lidam com entradas e saídas “binárias” e “reais” e armazenam “probabilidades contínuas” como conteúdos de memória. As RNSP com nodos pRAM apresentam características importantes, tais como: proximidade com os sistemas digitais convencionais; algoritmos de aprendizagem rápidos e flexíveis; precisão e consistência nos resultados, sem a necessidade de geração de resíduos e retreinamento das redes; nodos pRAM com entradas contínuas têm a capacidade natural de generalização (Clarkson *et. al.*, 1992); “redes” pRAM com entradas contínuas e aprendizado por reforço apresentam habilidades de aproximadores universais de funções contínuas (Gorse *et al.*, 1997). Com essas características, as redes pRAM passaram a resolver problemas anteriormente inapropriados para modelos neurais sem pesos, propostos anteriormente.

Na literatura relacionada às RNSP Oliveira *et al.* (2017) propuseram um sistema de FDD baseado no dispositivo WiSARD para o problema de detecção e diagnóstico de falhas em sistemas dinâmicos, descrito na seção 5.2.1 deste trabalho. A investigação com outros modelos de “neurônios” e “redes” baseados em memórias RAM para o problema de detecção e diagnóstico de falhas motivou o desenvolvimento deste trabalho. Assim, o seu objetivo é propor um sistema de detecção e diagnóstico de falhas baseado em redes neurais sem pesos probabilísticas para ambientes dinâmicos multivariáveis. O sistema proposto contou também com a contribuição da seleção de atributos e da inferência *fuzzy* para a formação dos padrões de entradas para as RNSP usadas. O referido sistema é validado por dois estudos de caso com dados simulados. Um deles é uma planta industrial considerada como um *benchmark* na área de detecção e diagnóstico de falhas e conhecida como *Tennessee Eastman Process* (TEP) (Downs e Vogel, 1993; Ricker, 1995). O outro estudo de caso simula um reator tanque agitado continuamente (CSTR, *Continuous Stirred Tank Reactor*) não isotérmico (Singhal e Seborg, 2002).

Na continuidade, a seção 5.2 apresenta os fundamentos sobre redes neurais sem pesos. Já a seção 5.3 apresenta o sistema FDD-pRAM (*Fault Detection and Diagnosis with Probabilistic RAM*, Detecção e Diagnóstico de Falhas com RAM Probabilísticas) e os métodos utilizados para os treinamentos e testes realizados. Na seção 5.4 são apresentados os dois estudos de caso utilizados com os seus respectivos resultados. Por último, na seção 5.5, são apresentadas as conclusões sobre os sistemas FDD-pRAM propostos para os estudos de caso considerados.

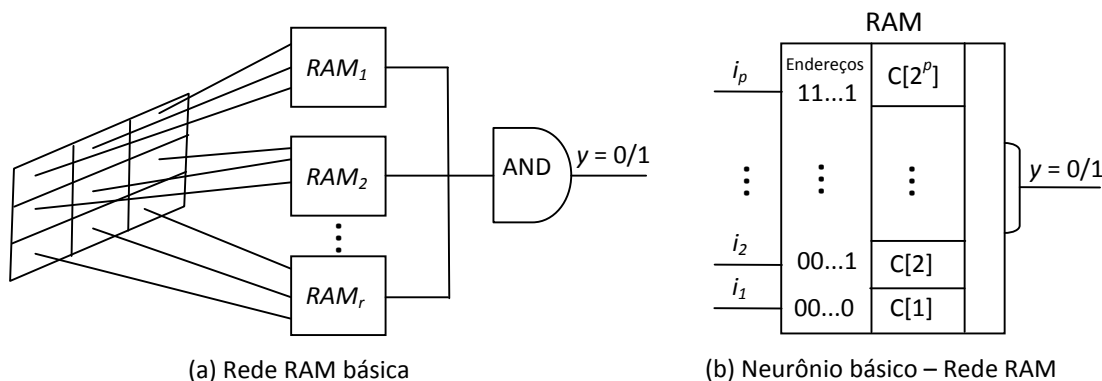
## 5.2 Redes Neurais sem Pesos

As Redes Neurais Artificiais (RNA), inspiradas inicialmente na funcionalidade dos neurônios biológicos, são formadas por pequenas unidades de processamento simples, dispostas em uma ou mais camadas com processamento paralelo e distribuído. As unidades de processamento recebem o nome de nodos ou neurônios artificiais, e a estrutura formada por essas unidades e suas conexões recebe o nome de rede neural artificial (Hecht-Nielsen, 1990). Com o processo de aprendizagem, as RNA são capazes de realizar atividades ditas “inteligentes”, mesmo que os dados utilizados nesse treinamento sejam não lineares, incompletos ou, até mesmo, contraditórios. Os algoritmos usados no processo de aprendizagem variam conforme o modelo de rede neural usado, mas são essencialmente derivados de dois neurônios básicos: o modelo de *McCulloch-Pitts* (MCP) (McCulloch e Pitts, 1943), inspirado na funcionalidade do neurônio natural e conhecido como neurônio “com pesos”, em função da ponderação que ele faz sobre as suas entradas; e o modelo digital, inicialmente chamado de SLAM (*Stored Logic Adaptive Microcircuit*) (Bledsoe e Browning, 1959) e posteriormente conhecido como RAM (*Random Access Memories*) (Aleksander, 1966), que deu origem a uma classe de neurônios baseados na lógica booleana, referenciados como neurônios “sem pesos” (*weightless neurons*). Na literatura

os termos *RAM-based* e *N-tuple-based* também são usados para referir-se às “Redes Neurais Sem Pesos” (RNSP) ou neurônios baseados em RAM (*RAM-based neurons*). Nas seções 5.2.1 a 5.2.3 são descritos os modelos neurais sem pesos necessários para a compreensão da rede *RAM-based* probabilística descrita na seção 5.2.4 e usada neste trabalho. Uma revisão desses modelos também pode ser encontrada em Ludermir *et al.*, (1999).

### 5.2.1 O Modelo RAM

O primeiro modelo de RNSP é a “rede neural RAM” [Figura 5.1(a)], cujos neurônios são nodos RAM [Figura 5.1(b)] (Oliveira *et al.*, 2017). Os nodos RAM recebem as entradas binárias que podem ser entradas externas, saídas de outros nodos ou de realimentação (conexões de *feedback*). O aprendizado em um nodo RAM consiste em modificações nos conteúdos de memória, armazenados em uma tabela-verdade. Para cada endereço  $a \in A = \{0, 1\}^p$  deve existir uma célula  $C[a]$ , onde são armazenados os conteúdos ou informações aprendidas (“memória local”) na forma de um  $q$ -bit, com  $q \in \{0, 1\}$  e inicializados por 0. Um padrão binário  $i \in \{0, 1\}^p$ , com  $i = i_1 i_2 \dots i_p$ , formado pelos  $p$  terminais de entradas  $i_1, i_2, \dots, i_p$ , pode acessar somente uma destas locações. O bit  $C[a]$  armazenado na memória ativada  $a = i$  representa a saída  $y$  de um nodo RAM, com  $y = C[a]$ . No nodo RAM, a resposta apresentada por qualquer posição endereçada tem a mesma importância, independentemente de ela ter sido treinada ou não. Assim, o estado de ativação de cada célula ao final do treinamento depende unicamente do último acesso a essa posição.

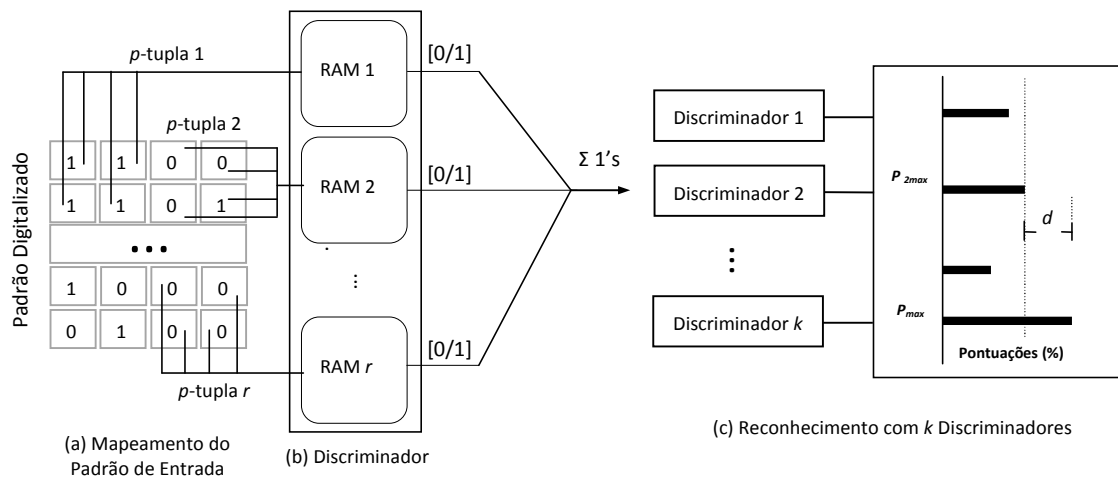


**Figura 5.1:** Estrutura da Rede e do Neurônio RAM (Oliveira *et al.*, 2017).

Uma rede RAM básica é treinada para reconhecer padrões de uma única classe, ou seja, padrões vistos no treinamento (classe 0). Um padrão de teste é reconhecido como da classe 0 somente se a saída de todas as RAM for igual a 1. Caso contrário, o padrão é posto na classe 1 e dito ser diferente dos padrões treinados (Oliveira *et al.*, 2017). Se, no problema em estudo, duas ou mais categorias são requeridas, utiliza-se uma estrutura com duas ou mais redes RAM. Cada rede RAM passa a configurar como um “discriminador” [Figura 5.2(b)] e treinada com padrões de uma única classe. Essa estrutura multidiscriminadora é amplamente usada e conhecida como dispositivo/ rede WISARD (Wilkie, Stonham e Aleksander’s Recognition Device, dispositivo de reconhecimento de Wilkie, Stonham e Aleksander) (Aleksander *et al.*, 1984).

A rede WiSARD é formada por uma única camada *feedforward* de discriminadores, compostos por nodos RAM. Esta rede é referida como clássica e usada principalmente como discriminadora de padrões. As RAM nos discriminadores são organizadas de modo que cada *bit* do padrão de entrada da rede seja associado a uma única RAM por meio de um mapeamento biunívoco pré-definido [Figura 5.2(a)]. Em comparação com a rede RAM, os discriminadores

são modificados de maneira que, em vez de ter uma porta lógica como saídas para combinar os nodos RAM, a decisão é levada para um somador ( $\Sigma$  1's), como mostrado na Figura 5.2(b). O valor do somador apresentado por um discriminador corresponde ao “fator de similaridade” do padrão de entrada com a classe relacionada a esse discriminador. O padrão de teste é reconhecido pelo discriminador com “maior” fator de similaridade, ou seja, maior número de RAM com saída igual a 1, conforme mostrado na Figura 5.2(c). O nível de confiança ( $C_w$ ) para o reconhecimento de um padrão  $w$  é calculado pela fórmula  $C_w = (P_{max} - P_{2,max}) / P_{max}$ , sendo  $P_{max}$  a pontuação máxima efetivamente obtida por um dos discriminadores e  $P_{2,max}$  a segunda maior pontuação (Ludermir *et al.*, 1999). O critério de “maior”, e não “todas” como na RAM básica, permite aos discriminadores da rede WiSARD reconhecerem padrões não vistos durante o treinamento, garantindo assim o poder de generalização da rede WiSARD.



**Figura 5.2:** Representação Esquemática da Rede WiSARD [Adaptado de Oliveira *et al.* (2017)].

O sucesso da generalização na rede WiSARD depende não somente da similaridade dos padrões de teste e treinamento, mas também da relação entre a quantidade e conectividade (valor de  $p$ ) das RAM nos discriminadores. Nessa relação, as grandezas quantidade e conectividade são inversamente proporcionais. Na WiSARD padrão, se todas as  $r$  RAM de um discriminador forem indexadas por  $RAM_i$ , com  $i$  inteiro  $\in [0, r-1]$ , o número de entradas  $p_i$  em todas as RAM para um padrão de entrada para a rede com  $s$  bits é definido pela Equação 5.1:

$$p_i = \begin{cases} \frac{s}{r} & \text{se } i < s \bmod r \\ r & \text{se } i \geq s \bmod r \end{cases} \quad (5.1)$$

Além disso, a quantidade de memórias nas RAM é função exponencial das entradas  $p$ , isto é, um nodo RAM tem  $2^p$  endereços de memórias. Um valor elevado de  $p$  requer uma quantidade maior de memória, provocando um espalhamento dos endereços acessados (Ludermir *et al.*, 1999). Esse espalhamento faz com que as RAM apresentem com maior frequência saídas iguais a 0 (zero), caso o padrão de teste não seja extremamente parecido com algum padrão aprendido pelo discriminador. Esse comportamento contribui para diminuir o fator de similaridade do padrão testado com a classe do discriminador referido. Por outro lado, um valor pequeno de  $p$  com muitos dados de treinamento provoca facilmente a saturação dos endereços de memória para os nodos RAM. A saturação acontece quando todos ou quase todos os endereços têm o seu conteúdo alterado para 1 (um). Consequentemente, as RAM passam a apresentar com maior frequência saídas iguais a 1 (um) e empates nas pontuações máximas apresentadas pelos

discriminadores. Neste caso, a rede WiSARD padrão utiliza o critério da “aleatoriedade” para definir um dos discriminadores empatados como vencedor. Porém, o uso da aleatoriedade diminui a acurácia da rede, pois o padrão testado tem grande possibilidade de ser classificado erroneamente.

Os efeitos negativos com a “aleatoriedade” na rede RAM podem ser amenizados com o uso da técnica de *bleaching* (refinamento) proposta por Aleksander e Morton (1990). A técnica de *bleaching* consiste em usar um valor de limiar  $b$ , com  $b \leq 1$ , para realizar um desempate mais confiável entre os discriminadores de uma rede WiSARD. Para isso, os endereços de memória nas RAM passam a armazenar um contador de valor inteiro, ao invés dos valores 0 e 1. Desse modo, a saída de uma RAM é igual a 1 se o valor inteiro armazenado for igual ou superior ao valor do limiar e 0 em casos contrários. Na prática, a técnica de *bleaching* (limiar) pode ser implementada de modo “homogêneo” ou “heterogêneo”.

No *bleaching* homogêneo o mesmo valor de limiar é usado para todos os discriminadores. A cada verificação, se o conteúdo acessado é maior do que o limiar, a saída da RAM é 1. Porém, se dois ou mais discriminadores apresentarem o mesmo somatório de valores 1 o valor do limiar é acrescido de uma unidade. Com isso os conteúdos das RAM serão novamente verificados de modo a reduzir o número de RAM com saída igual a 1. Este procedimento é repetido até que um discriminador seja eleito ou até que todos parem de pontuar, situação em que a escolha é feita de modo aleatório.

No *bleaching* heterogêneo o valor do limiar varia de acordo com a quantidade de aprendizado em cada discriminador. Dessa forma evita-se que uma categoria com mais treinamentos seja “privilegiada”, pois os limiares heterogêneos têm a mesma representatividade e criam um equilíbrio entre os discriminadores (Cardoso *et al.*, 2016; Carneiro *et al.*, 2015). Para isso utiliza-se o *bleaching* convencional e o número de treinamentos realizados por cada discriminador para determinar o grau de importância de um padrão testado, sendo a saída  $y$  de um RAM obtida com a Equação 5.2:

$$y = \begin{cases} 1 & \text{se } \frac{C[a]}{nt(D_i)} > \frac{\text{Bleaching convencional}(\cdot)}{nt(D_{hf})} \\ 0 & \text{caso contrário} \end{cases} \quad (5.2)$$

onde  $C[a]$  é o conteúdo do endereço  $a$  acessado pelo padrão de entrada  $I$ ,  $nt(D_i)$  é o número de treinamentos do discriminador  $i$  contendo  $C[a]$ , com  $1 \leq i \leq k$  ( $k$  é quantidade de discriminadores com  $r$  RAM cada),  $\cdot = 0, 1, 2, \dots$  são os possíveis valores para o *bleaching* convencional e  $nt(D_{hf})$  é o número de treinamentos realizado pelo discriminador referente à classe com maior número de padrões.

O dispositivo WiSARD e a técnica de *bleaching* contribuíram significativamente para o avanço nas pesquisas com RNSP. No entanto, outros modelos de neurônios/ nodos sem pesos foram propostos, com destaque para os modelos probabilísticos: PLN (*Probabilistic Logic Neuron*); MPLN (*Multiple-valued Probabilistic Logic Neuron*); e pRAM (*probabilistic RAM*); descritos, respectivamente, nas seções 5.2.2 a 5.2.4 seguintes.

### 5.2.2 O Modelo PLN

O modelo de neurônio PLN é uma versão probabilística do nodo RAM, proposto por Kan e Aleksander, depois de tentativas de aproximação entre a máquina de Boltzman e redes RAM (Kan e Aleksander, 1987). O propósito do modelo é saber diferenciar se uma posição de memória foi treinada ou não. Para isso, todos os endereços de memória são inicializados por um valor indefinido  $u$  e no aprendizado os endereços acessados são alterados para 0 ou 1. Na

fase de teste, se um endereço com valor  $u$  é acessado, pode-se afirmar que não houve treinamento nesse endereço. Neste caso, o nodo PLN produz uma resposta 0 ou 1 com a mesma probabilidade. Assim, a função de saída  $y$  do nodo PLN é definida pela Equação 5.3:

$$y = \begin{cases} 0 & \text{se } C[a] = 0 \\ 1 & \text{se } C[a] = 1 \\ \text{random}(0,1) & \text{se } C[a] = u \end{cases} \quad (5.3)$$

onde  $C[a]$  é o conteúdo acessado pelo padrão de entrada  $a$  e a função aleatória  $\text{random}(0,1)$  produz uma saída 0 ou 1 com a mesma probabilidade.

Embora haja flexibilidade quanto à arquitetura de uma rede neural composta por nodos PLN, este tipo de rede geralmente utiliza a estrutura piramidal (Aleksander, 1989). A pirâmide é uma estrutura em forma de árvore na qual cada camada tem um número fixo de neurônios com uma única saída (*fan-out* igual a 1; *fan-out*= $n^\circ$  de saídas). Uma justificativa para o uso da arquitetura piramidal é a redução da dimensionalidade do espaço de entradas (*fan-in* baixo; *fan-in*= $n^\circ$  de entradas), economizando memória nos neurônios construtores da rede.

### 5.2.3 O Modelo MPLN

O nodo PLN multivalorado, chamado de MPLN (*Multiple-valued Probabilistic Logic Neuron*) (Myers e Aleksander, 1988, 1989), foi motivado pelo uso do terceiro valor lógico usado no PLN. O MPLN armazena a probabilidade de o neurônio responder com 1 quando um endereço é acessado, ao invés de armazenar um elemento do conjunto (0, 1,  $u$ ). Os conteúdos de memória no nodo MPLN são inicializados com 0,5, o equivalente ao valor  $u$  do PLN. Com o processo de aprendizagem, os conteúdos de memória são incrementados ou decrementados, podendo chegar a 0 (limite inferior) ou a 1 (limite superior) (Myers e Aleksander, 1988). A função de ativação para o nodo MPLN aplicada ao conteúdo endereçado é definida pela Equação 5.4:

$$y = \begin{cases} C[a] - \eta g(r) & \text{punição} \\ C[a] + \eta g(r) & \text{recompensa} \end{cases} \quad (5.4)$$

onde  $C[a]$  é o conteúdo de memória endereçado pelo padrão de entrada  $a$ ,  $\eta$  é a taxa de aprendizado e  $g(r)$  é definida pela Equação 5.5:

$$g(r) = \begin{cases} +1 & \text{se } r = 1 \\ -1 & \text{se } r = 0 \end{cases} \quad (5.5)$$

A rapidez com que cada valor chega a 0 ou a 1 depende do número de elementos definidos no conjunto de possíveis valores armazenáveis. Como exemplo, é esperado que os conteúdos do conjunto (0; 0,5; 1), como no PLN, cheguem primeiro aos extremos 0 e 1 do que o conjunto (0; 0,2; 0,4; 0,6; 0,8; 1). A maneira como o conteúdo inicial é modificado para chegar a 0 ou 1 consiste na estratégia de punição ou recompensa usada na fase de aprendizagem. Assim, uma simples punição ou recompensa não é suficiente para definir ou apagar completamente um comportamento, pois a mudança ocorre de forma gradativa e mediante os vários exemplos de treinamento (Myers e Aleksander, 1988).

O tempo de convergência para um nodo MPLN disparar uma saída 0 ou 1 depende do número de *bits* usados na representação das probabilidades armazenadas nas locações de memória do nodo. Se esse número for muito grande, a convergência implica num período de tempo maior e num aprendizado mais lento para um conjunto de treinamento com muitos exemplos. Uma boa estimativa no número de *bits* para a representação dessas probabilidades depende da especificidade do problema. O limite natural do nodo MPLN é determinado por “probabilidades

contínuas” com valores  $\in [0, 1]$ , endereçados por  $p$  bits. Esta condição define o nodo “RAM probabilístico” de Gorse e Taylor, denominado de pRAM (*probabilistic RAM*) (Gorse e Taylor, 1988) e descrito na seção seguinte.

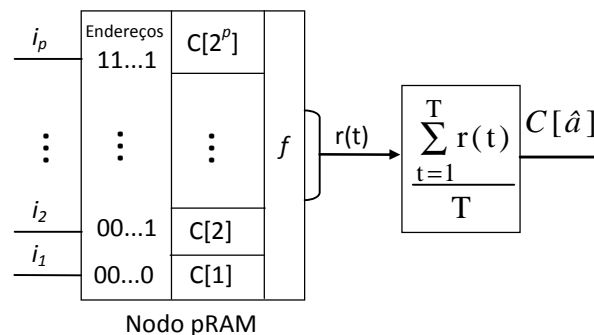
### 5.2.4 Modelo pRAM

O nodo RAM probabilístico pRAM, assim como o MPLN, é também considerado como uma extensão do PLN e foi desenvolvido simultaneamente ao MPLN e de forma independente. O que difere o pRAM do PLN é a característica de poder armazenar em suas locações de memória probabilidades “contínuas” com valores num intervalo de  $[0, 1]$ . Essas probabilidades dão mais realismo ao modelo, visto que o “ruído”, presente em todos os sistemas reais, pode ser simulado no nodo pRAM. As entradas e saídas dos nodos pRAM podem ser implementadas com valores binários ou contínuos. O modelo pRAM, originalmente foi proposto no domínio binário, sendo depois estendido para mapear entradas contínuas com saídas binárias. Esta extensão foi chamada de “*integrating pRAM*” (*i-pRAM*) (Gorse e Taylor, 1991). O *i-pRAM* tem a capacidade natural de generalização e, conseqüentemente, permite aos nodos pRAM resolverem problemas anteriormente inapropriados para os outros modelos neurais RAM-based.

Um nodo pRAM com entradas binárias e saídas real possui  $2^p$  possíveis locações de memória endereçáveis por um vetor de entrada  $a \in A = \{0,1\}^p$ , conforme mostrado na Figura 5.3 [originalmente apresentada em Gorse e Taylor (1988)]. Um sinal binário  $i = i_1 i_2 \dots i_p$  nas linhas de entradas pode acessar somente uma destas locações (*i.e.*,  $a = i$ ). O conteúdo  $C[a]$  armazenado, até que esta locação de memória seja ativada, representa a probabilidade que assegura um valor igual a 1 ser produzido na linha de saída  $y$ , com base em uma entrada conhecida  $i$ . Contudo, no caso de entradas binárias, o modo de obter o valor  $y$  na saída do nodo pRAM é conhecido pelo conteúdo da posição de memória  $C[a = i]$  que está sendo acessada. Portanto, quanto maior o valor dos conteúdos das posições de memória, maior a probabilidade de a saída do nodo ser igual a 1, conforme Equação 5.6:

$$\begin{aligned}
 y &= \sum_{a \in A} C[a] \delta a, i \\
 &\equiv C[a = i] = C[\hat{a}]
 \end{aligned}
 \tag{5.6}$$

onde  $\delta a, i$  é a função que determina o endereço  $a$  a partir da entrada  $i$ .



**Figura 5.3:** Processo para Computar a Saída de um Nodo pRAM com Entradas Binárias e Saída Real [Adaptado de Ludermir *et al.* (1999)].

O nodo pRAM com entradas contínuas, como mostrado na Figura 5.4 [originalmente apresentada em Gorse e Taylor (1991)], tem suas entradas reais  $x \in [0, 1]^p$  aproximadas por

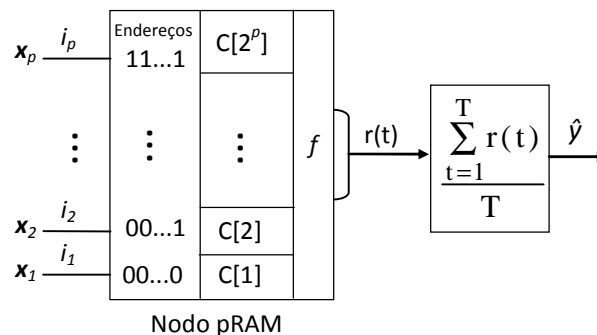


algum período de tempo,  $T$ , determinado por sucessivas entradas de padrões binários  $i \in \{0,1\}^p$ , com  $i = i_1 i_2 \dots i_p$ . Neste caso cada  $x_j (j = 1, 2, \dots, p)$  poderá ser representado por uma sequência de sinais  $i_j$ , sendo que  $x_j$  representa a probabilidade de ocorrer uma saída 1 na  $j$ -ésima posição de memória de  $i$  (i.e.,  $P(i_j = 1) = x_j$ ). Ao contrário do pRAM original com entradas binárias ou qualquer outro nodo sem pesos, no nodo pRAM com entradas contínuas a identificação da probabilidade de saída associada com a entrada atual não é feita por uma única entrada específica na tabela-verdade, mas sim através de um espaço de tempo,  $t = 1, \dots, T$ , necessário para acessar as locações de memória que irão “contribuir” estocasticamente para produzir um valor de saída 1 ou 0 para uma sequência de saídas. Nestas condições a saída  $y$  do nodo pRAM com entradas contínuas é dada pela equação 5.7:

$$y = \sum_a C[a] \prod_{j=1}^p [a_j x_j + (1-a_j)(1-x_j)]$$

$$y \cong \sum_a C[a] P_a(x)$$
(5.7)

onde  $C[a]$  representa o conteúdo da memória de endereço  $a$  e  $P_a(x)$  é a probabilidade desse endereço  $a$  ser acessado. Desse modo, uma probabilidade é computada com base na função de distribuição de probabilidade definida por  $x$ . Esta função de saída pode ser vista como um produto de polinômios construído a partir dos  $p$  sinais de entrada para as RAM.



**Figura 5.4:** Processo para Computar a Saída de um Nodo pRAM com Entradas Contínuas e Saída Real [Adaptado de Ludermir *et al.* (1999)].

No caso de implementação em *hardware* (Clarkson *et al.*, 1992), a saída do pRAM é aproximada para um período de tempo  $T$  de sucessivos *bits* de saída (fluxo de saída), independentemente de entradas contínuas ou binárias e definida conforme a Equação 5.8:

$$y = \frac{1}{T} \sum_{t=1}^T r(t)$$
(5.8)

Se  $T = 1$  e  $P_a(x) \in \{0,1\}$ , o nodo pRAM equivale ao nodo MPLN. Com  $T > 1$  e  $P_a(x) \in [0, 1]$ , o nodo pRAM é muito mais complexo, e esta é a principal diferença para o nodo MPLN (Ludermir *et al.*, 1999). Além disso, os nodos pRAM com entradas contínuas apresentam características tais como (Gorse *et al.*, 1997): (1) têm propriedades de generalizações; (2) não devem ser considerados como tabelas-verdade, do mesmo modo que nos nodos sem pesos convencionais, visto que a saída do nodo pRAM poderá conter um valor obtido por um extenso intervalo de locações.

Como no modelo PLN, os nodos MPLN e pRAM têm sido usados em arquiteturas piramidais com aprendizado por reforço. No caso do pRAM, o aprendizado por reforço é definido pela Equação 5.9:

$$C[a] = [(a - C[a]) + (\beta - C[a])] \quad (5.9)$$

onde  $\beta$  e  $\alpha$  são a recompensa e punição associadas a uma tarefa de entrada-saída,  $a$  é a probabilidade de uma saída 0 (zero) ou 1 (um) do nodo pRAM num tempo atual,  $t$ ,  $\beta = 1 - \alpha$ ,  $\alpha$  é uma taxa de formação de recompensa e  $\beta$  é a taxa de formação de punição associada. A transformação  $C[a] = \alpha C[a] + \beta$  pode ser implementada diretamente por um nodo pRAM com entradas  $a$ , uma recompensa  $\beta$  e um conteúdo de memória  $C[a]$ , com  $\alpha = 1 - \beta$ . As redes pRAM com aprendizado por reforço apresentam habilidades de aproximadores universais de funções contínuas (Gorse *et al.*, 1997).

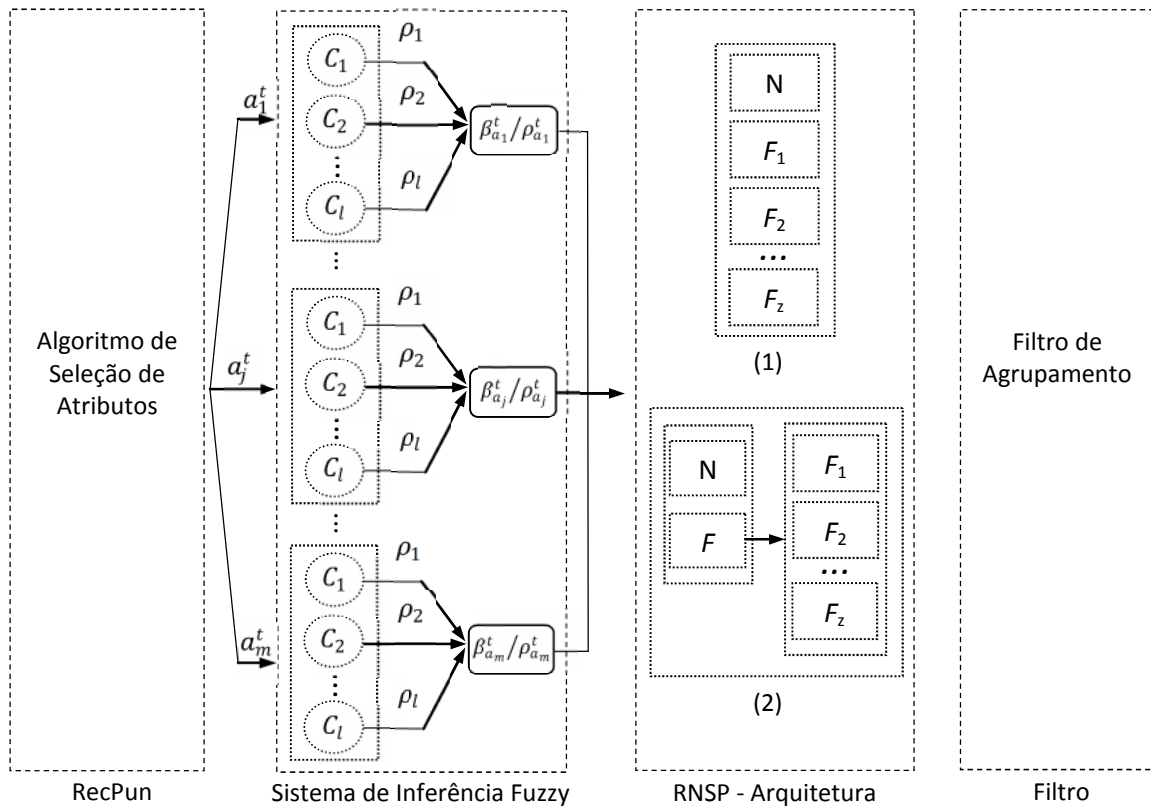
No sistema FDD aqui proposto as RNSP são compostas por nodos pRAM com entradas contínuas e saídas binárias e apresentam arquiteturas baseadas no dispositivo WiSARD com o uso da técnica de *bleaching* (Aleksander *et al.*, 1984; Aleksander e Morton, 1990). Cada discriminador na WiSARD é treinado com os atributos selecionados para a classe relativa a cada discriminador, com base no algoritmo de seleção de atributos descrito na seção 5.3.2.

### 5.3 O Sistema Neural sem Pesos Probabilístico FDD-pRAM Proposto

#### 5.3.1 Estrutura do Sistema

O sistema proposto neste trabalho, denominado de FDD-pRAM, realiza a detecção e o diagnóstico de falhas em dados multivariáveis (vetores  $n$ -dimensionais). Cada dimensão corresponde a uma variável (atributo) obtida por um simulador ou através de dados históricos. Em termos estruturais o sistema é composto por quatro etapas distintas e complementares, denominadas de (Figura 5.5): “Algoritmo de Seleção de Atributos – RecPun”; “Sistema de Inferência *Fuzzy*”; “Rede Neural Sem Pesos – RNSP”; e “Filtro de Agrupamentos – Filtro”. Inicialmente o sistema FDD utiliza o algoritmo de seleção de atributos por classe, proposto por Vale *et al.* (2010), com algumas modificações e adaptado para dados multivariáveis. O sistema de inferência *fuzzy* utiliza o conceito de *clouds* proposto por Angelov e Yager (2011) para a granulação dos atributos selecionados, e aqui adaptado para lidar com dados multivariáveis. As redes neurais sem pesos são constituídas por nodos pRAM com entradas contínuas e saídas binárias, e realizam a detecção e o diagnóstico de falhas de forma paralela ou em cascata. As camadas da rede são arquiteturas baseadas na rede WiSARD (Aleksander *et al.*, 1984). Opcionalmente, um diagnóstico por grupo de falhas pode ser obtido por um filtro de agrupamento como etapa final do sistema.

No contexto dos sistemas de FDD, a “detecção” consiste em determinar a presença de um comportamento indesejável ou inaceitável do sistema. Já o “diagnóstico” compreende no isolamento (classificação, localização e tempo de ocorrência das falhas) e identificação/ análise (tamanho, causa e comportamento temporal, origem e impactos no sistema) das falhas detectadas (Isermann e Ballé, 1997). Como o sistema de FDD aqui proposto realiza apenas a detecção e classificação dos padrões de entrada, ele pode também ser referenciado como um sistema de detecção e isolamento de falhas (FDI - *Fault Detection and Isolation*).



**Figura 5.5:** Estrutura do Sistema Neural sem Pesos: FDD-pRAM.

No sistema da Figura 5.5 a detecção e o diagnóstico de falhas acontecem por meio do reconhecimento e classificação de padrões, definidos por uma aproximação para a relação  $y = f(x)$ , sendo  $x$  um vetor  $n$ -dimensional ( $n$  atributos) de valores reais e  $y$  uma variável de resposta. Nessa relação  $y$  é um rótulo de classe com valor no conjunto  $\{C_1, C_2, \dots, C_k\}$ , com  $k \in \mathbb{N}$  e a função  $f(x)$  define, de forma aproximada, os limites para as classes consideradas. Num contexto temporal, e considerando apenas a detecção de falhas, os atributos  $a_1^t, a_2^t, \dots, a_n^t$  deverão conter informações necessárias para classificar o padrão  $x = [a_1^t, a_2^t, \dots, a_n^t]$  como estado de operação “normal” ou estado de “falha” no instante  $t$ . A partição do estado de falha em subclasses com rótulos definidos pelos diferentes tipos ou grupos de falhas considerados compreende o diagnóstico de falhas realizado pelo sistema. Para melhor compreensão, as etapas do sistema FDD-pRAM são apresentadas com mais detalhes nas seções 5.3.2 a 5.3.5.

### 5.3.2 Seleção de Atributos

A qualidade da “generalização” num sistema de detecção e diagnóstico de falhas com aprendizado supervisionado depende da “distinção” dos limites de classes obtida pelos classificadores envolvidos (Kuncheva e Whitaker, 2003). Em aplicações multivariáveis, a seleção de atributos pode contribuir para a melhoria nessa distinção dos limites de classes pelos classificadores do sistema (Tsymbal *et al.*, 2003). Com esse propósito, este trabalho utiliza o algoritmo de seleção de atributos por classe denominado de RecPun (RECompensa/ PUNição), proposto por Vale *et al.* (2010). Esse algoritmo usa a recompensa e a punição para ordenar os atributos de acordo com as suas importâncias para cada uma das classes existentes. A ideia do modelo é que exista, no mínimo, um classificador por classe e que os atributos selecionados para uma dada classe sejam importantes somente para a referida classe (Vale *et al.*, 2010).

Porém, ao aplicar o algoritmo RecPun aos estudos de caso considerados neste trabalho, os atributos selecionados foram os mesmos para a maioria das classes. Para amenizar essa situação foi incluído um passo adicional ao final ao algoritmo original. O passo acrescido determina que, numa comparação biunívoca entre as classes consideradas, deva existir, no máximo, 50% de atributos comuns (método aqui denominado de RecPun modificado). Nessa comparação biunívoca, se a quantidade de atributos comuns de uma dada classe  $i$ , com  $i = 1, 2, \dots, k$ , comparados com os atributos das classes anteriores for superior a 50%, mantêm-se os primeiros 50% e os demais são substituídos por atributos de menor importância. No entanto, se a quantidade de atributos não for suficiente para a substituição dos demais atributos, a ordenação da segunda metade dos atributos da classe comparada é modificada com base no critério da aleatoriedade. Um estudo detalhado do algoritmo RecPun original pode ser encontrado em Vale *et al.* (2010) e uma versão simplificada em Oliveira *et al.* (2017).

### 5.3.3 Sistema de Inferência Fuzzy para os Atributos Selecionados

O sistema de inferência *fuzzy* na Figura 5.5 utiliza o conceito de *clouds* proposto em Angelov e Yager (2011) para realizar a granulação do espaço de entrada. O trabalho aqui proposto tem como base o algoritmo apresentado por Rosa *et al.* (2013) com as adaptações para lidar com dados multivariáveis. Para Angelov e Yager (2011) as *clouds* criam uma partição dos dados de entrada semelhante a uma estrutura de grupo, porém não há uma formação explícita de grupos. No algoritmo proposto por Rosa *et al.* (2013) as *clouds* são criadas ou atualizadas de maneira recursiva e com base em densidades locais e global obtidas com o processamento do fluxo de dados no espaço de entrada. Nas aplicações consideradas neste trabalho as variáveis (atributos) são consideradas fluxos de dados e processadas de forma paralela.

Como mostrado na Figura 5.5, o processo de granulação acontece após a etapa de seleção de atributos (variáveis) e cria uma estrutura de *clouds* para cada atributo (variável) selecionado. No restante desta seção é apresentado o algoritmo utilizado por Rosa *et al.* (2013) com as modificações necessárias para lidar com dados multivariáveis. Assim, a densidade local  $\gamma_{ij}^t$  do padrão de entrada  $x^t = [a_1^t, a_2^t, \dots, a_m^t]$  em  $t$ , para a  $i$ -ésima *cloud* e atributo  $j$ , com  $i = 1, 2, \dots, l$  e  $j=1, 2, \dots, m$ , é calculada recursivamente pela Equação 5.10:

$$\gamma_{ij}^t = \frac{1}{1 + \left\| a_j^t - \mu_{tj}^t \right\|^2 + \Sigma_{ij}^t - \left\| a_j^t \right\|^2}, \quad (5.10)$$

com  $\mu_{ij}^t = ((M_{ij}^t - 1) / (M_{ij}^t)) M_{ij}^{t+1} + a_j^t / (M_{ij}^t)$  e  $\mu_{1j}^1 = a_j^1$ , sendo  $\mu_{ij}^t$  o valor modal local,  $M_{ij}^t$  é o número de dados de entrada associados à *cloud*  $i$  e  $\Sigma_{ij}^t$  é a dispersão local definida pela equação 5.11:

$$\Sigma_{ij}^t = \frac{M_{ij}^t - 1}{M_{ij}^t} \Sigma_{ij}^{t-1} + \frac{1}{M_{ij}^t} \left\| a_j^t \right\|^2, \quad \Sigma_{1j}^1 = \left\| a_j^1 \right\|^2 \quad (5.11)$$

Semelhantemente à densidade local, a densidade global também é calculada de forma recursiva e com base no total de dados de entrada. Consequentemente, o valor modal e a dispersão são modificados para valores globais. Assim, Equação 5.12 define recursivamente a densidade global  $\Gamma_j^t$  em  $t$  para o atributo  $j$ , com  $j=1, 2, \dots, m$ :

$$\Gamma_j^t = \frac{1}{1 + \left\| a_j^t - \mu_{G_j}^t \right\|^2 + \sum_{G_j}^t - \left\| a_j^t \right\|^2} \quad (5.12)$$

com  $\mu_{G_j}^t = ((t-1)/(t))\mu_{G_j}^{t-1} + a_j^t/(t)$  e  $\mu_G^1 = a_j^1$ , sendo  $\mu_{G_j}^t$  o valor modal global para o total de dados até o tempo  $t$ , cuja dispersão global é definida pela Equação 5.13:

$$\sum_{G_j}^t = \frac{t-1}{t} \sum_{G_j}^{t-1} + \frac{1}{t} \left\| a_j^t \right\|^2, \quad \sum_{G_j}^1 = \left\| a_j^1 \right\|^2 \quad (5.13)$$

Além das densidades locais e global, o procedimento de criação ou atualização de uma *cloud* depende do valor do ponto focal,  $\beta_{i_j}^f$ . O ponto focal corresponde ao dado de entrada com maior densidade local e global da *cloud*  $i$  e atributo  $j$ , com  $i = 1, \dots, l$  e  $j = 1, 2, \dots, m$ . Em termos práticos, o dado (atributo) de entrada  $a_j^1$  define a primeira *cloud* e seu respectivo ponto focal, isto é,  $\beta_{1_j}^f \leftarrow a_j^1$ . Na continuidade do fluxo de dados uma *cloud* é criada se a densidade global de um dado de entrada é maior do que a densidade global de cada *cloud* no seu ponto focal, isto é se  $\Gamma_j^t > \Gamma_{i_j}^f \forall i, j$ , sendo  $\Gamma_{i_j}^f$  a densidade global do ponto focal da *cloud*  $i$  para o atributo  $j$ . Caso contrário, atualiza-se o ponto focal da *cloud*  $i$  com maior densidade local para o dado (atributo) de entrada  $a_j^t$ , conforme a Equação 5.14:

$$cloud_{i_j} = \arg \max_i (\gamma_{i_j}^t) \quad (5.14)$$

Um dado (atributo) de entrada  $a_j^t$  é considerado o novo ponto focal de uma *cloud*  $i$  se a densidade global e a densidade local dessa *cloud*  $i$  forem maiores do que  $\Gamma_{i_j}^f$  e  $\gamma_{i_j}^f$ , respectivamente. Por último, define-se o grau de compatibilidade de um dado (atributo) de entrada  $a_j^t$  para as *clouds* existentes através da Equação 5.15:

$$\theta_{i_j}^t = \frac{\gamma_{i_j}^t}{\sum_{i=1}^l \gamma_{i_j}^t} \quad \forall j \quad (5.15)$$

onde  $\gamma_{i_j}^t$  é a densidade local da entrada  $a_j^t$  na *cloud*  $i$  para o atributo  $j$ .

Com base no algoritmo descrito acima, a cada padrão de entrada  $x^t = [a_1, a_2, \dots, a_m]$  são obtidos  $m$  pontos focais  $\beta^i$  e  $m$  graus de compatibilidade  $\theta^i$ , postos como saídas das *clouds* que acomodam  $a_1, a_2, \dots, a_m$ , respectivamente.

#### 5.3.4 Estrutura da Rede Neural sem Pesos

Conforme já mencionado anteriormente, a RNSP pode realizar a detecção e o diagnóstico de falhas de forma paralela ou em cascata, com base nas “duas arquiteturas” mostradas na etapa 3 da Figura 5.5 e descritas abaixo. Essas redes são compostas por uma ou duas camadas de discriminadores (classificadores) com a estrutura da rede WiSARD. Os discriminadores são

constituídos por neurônios lógicos probabilísticos pRAM de Gorse e Taylor (1988) com entradas contínuas e saídas binárias, descritos na seção 5.2.4.

- ✓ Arquitetura 1 [Figura 5.5 - RNSP(1)]. A RNSP é composta por uma “única camada”, contendo uma única WiSARD com  $k$  discriminadores treinados individualmente para representar uma única classe de padrões. Os estados de operação Normal ( $N$ ) e de Falhas ( $F_1, F_2, \dots, F_z$ ) são representados pelos discriminadores  $N, F_1, F_2, \dots, F_z$ , respectivamente. Na validação, o padrão testado é apresentado a todos os discriminadores e classificado como pertencente à classe do discriminador com maior somatório de valores 1 como saídas das RAM, conforme modelo apresentado na Figura 5.2.
- ✓ Arquitetura 2 [Figura 5.5 RNSP(2)]. A RNSP é composta por “duas camadas”. A primeira realiza a detecção de falhas por meio de uma WiSARD com dois discriminadores treinados individualmente para representar os padrões do estado de operação Normal ( $N$ ) e de Falha ( $F_1, F_2, \dots, F_z$ ). A segunda camada realiza o diagnóstico de falhas com uma WiSARD com  $z$  discriminadores treinados individualmente para representar os padrões de Falhas ( $F_1, F_2, \dots, F_z$ ). Na validação, se a primeira camada classificar o padrão testado como de falha, o teste é realizado também na segunda camada, de modo a identificar o tipo de falha do padrão testado.

Como os atributos são selecionados por classe, os discriminadores nas duas arquiteturas são estruturados para o aprendizado ou validação de padrões comportamentais, obtidos com o sistema de inferência *fuzzy* para os atributos selecionados. Um padrão comportamental é a concatenação dos  $m$  pontos focais,  $\beta^t = \beta_{a_1}^i \beta_{a_2}^i \dots \beta_{a_m}^i$ , representados em binário simples e formados com as saídas das  $m$  *clouds* vencedoras, como descrito no final da seção anterior. O padrão comportamental  $\beta^t$  corresponde à entrada da RNSP e, mais especificamente, às entradas para as RAM do discriminador representante da classe do padrão  $x^t$ . O tamanho de  $\beta^t$  em *bits* é  $m$  vezes o número de *bits* usados para representar  $\beta_{a_j}^t \forall j$  com  $j=1, 2, \dots, m$ .

Numa RNSP com estrutura baseada no dispositivo WiSARD, os discriminadores são treinados, individualmente, apenas com padrões positivos de uma única classe. Assim, os nodos pRAM usados no sistema FDD-pRAM são treinados com entradas contínuas e incrementos positivos nos conteúdos de memória, com base na Equação 5.7. No processo de treinamento os conteúdos de memória acessados por  $\beta^t$  são inicializados por 0 (zero) e atualizados com o valor definido pela função “*t*-norma produto” dos  $m$  graus de compatibilidade  $\rho_{a_j}^t$ , representada por  $\rho^t =$

$\rho_{a_1}^i \Lambda \rho_{a_2}^i \Lambda \dots \Lambda \rho_{a_m}^i$ , sendo  $\Lambda$  o operador da função *t*-norma produto (Zadeh, 1994). Assim, a

probabilidade de saída associada com a entrada atual  $x^t$  é obtida por um período de tempo,  $t = 1, \dots, T$ , necessário para acessar as locações de memória que irão “contribuir” estocasticamente para produzir uma saída binária 1 ou 0. De modo a tornar-se compatível com o procedimento usado na rede WiSARD, a probabilidade de saída do nodo pRAM aqui implementado é igual a 1 (um) se o conteúdo de memória  $C[a]$  for maior ou igual a um valor de *bleaching* definido. Caso contrário, a saída é 0 (zero). Neste trabalho o *bleaching*,  $b > 0$ , assume um valor real no intervalo  $[0, 1]$ , equiparando o nodo pRAM a um nodo RAM com *cut-point* (ponto de corte). Assim, um padrão testado é apresentado a todos os discriminadores e classificado como pertencente à classe do discriminador com maior somatório de valores 1 como saídas das RAM, como mostrado na Figura 5.2. Como os nodos pRAM armazenam probabilidades contínuas, a diferenciação entre elas só é possível com incrementos ínfimos no valor do *bleaching*. Com os

incrementos no valor do *bleaching*, o número de nodos pRAM com saída igual a 1 diminui de modo a definir um único discriminador como vencedor ou usar o critério da aleatoriedade com uma quantidade significativamente menor de discriminadores.

O nível de sensibilidade e especificidade do sistema para a detecção e o diagnóstico de falhas depende não somente dos incrementos feitos no valor *bleaching*, mas também da avaliação de outros parâmetros, tais como: i) quantidade de atributos selecionados (1, 2, ...,  $m$ ); ii) representação binária dos  $m$  pontos focais definidos no processo de inferência *fuzzy* para cada padrão processado (8, 12, 16 ou 32 *bits*); iii) tamanho dos padrões de entrada para a rede igual à multiplicação do item (ii) por  $m$  (número de atributos selecionados); iv) número de entradas para as RAM igual a 2, 3, 4, 6 ou 8 *bits* [Figura 5.1(b)]; v) número de pRAM por discriminadores igual à divisão exata do item (iii) pelo item (iv); vi) uso do *bleaching* simples ou percentual; vii) método de seleção de atributos (RecPun ou RecPun modificado).

### 5.3.5 Filtro de Agrupamentos

Conforme descrito anteriormente, a detecção e o diagnóstico de falhas são realizados de forma simultânea [Figura 5.5(1)] ou em cascata [Figura 5.5(2)]. Simultaneamente, a detecção e o diagnóstico são realizados numa única etapa e classificam os padrões de testes como estado de operação normal ou falha específica. Em cascata, a camada inicial da rede realiza a detecção de falhas, classificando os padrões de testes como estado de operação normal ou de falhas, enquanto na segunda e última camada da rede é realizado o diagnóstico dos padrões de falhas por tipo de falha. No entanto, nos dois modelos de arquitetura a saída final da rede classifica os padrões de testes como estado de operação normal ou falha específica. Porém, a partir das saídas da rede, se um diagnóstico por grupo de falhas com características comuns for desejado, um filtro de agrupamento é usado como etapa final do sistema FDD-pRAM. Assim, é possível uma análise dos resultados em nível da rede (diagnóstico mais granulado) ou do sistema como um todo, a exemplo dos resultados e análises apresentados nas seções 5.4.1.1 e 5.4.2.1.

### 5.3.6 Análise de Desempenho

Os resultados apresentados neste trabalho são expressos em taxas percentuais de acerto por classe/ grupo de padrões. Os resultados apresentados pelos sistemas estão dispostos em uma matriz de classificação com conteúdos variados. A exatidão do sistema na detecção e no diagnóstico de falhas foi utilizada como critério de escolha do sistema final para cada estudo de caso. Os percentuais de acertos/ erros referem-se às classificações devidas/ indevidas dos dados do conjunto de teste, com base nos limites estabelecidos para as classes consideradas. A exatidão dos sistemas, tanto para a detecção de falhas quanto para o diagnóstico, é obtida na saída do sistema como um todo (rede acoplada ao filtro de agrupamento). Conforme mostrado na Figura 5.5, o filtro aplicado na detecção de falhas classifica os padrões de teste como estado de operação normal ou estado de falha. Já o filtro usado para o diagnóstico das falhas considera os estados representados por operação normal ou estado de falha do tipo  $i$  ou grupo de falhas.

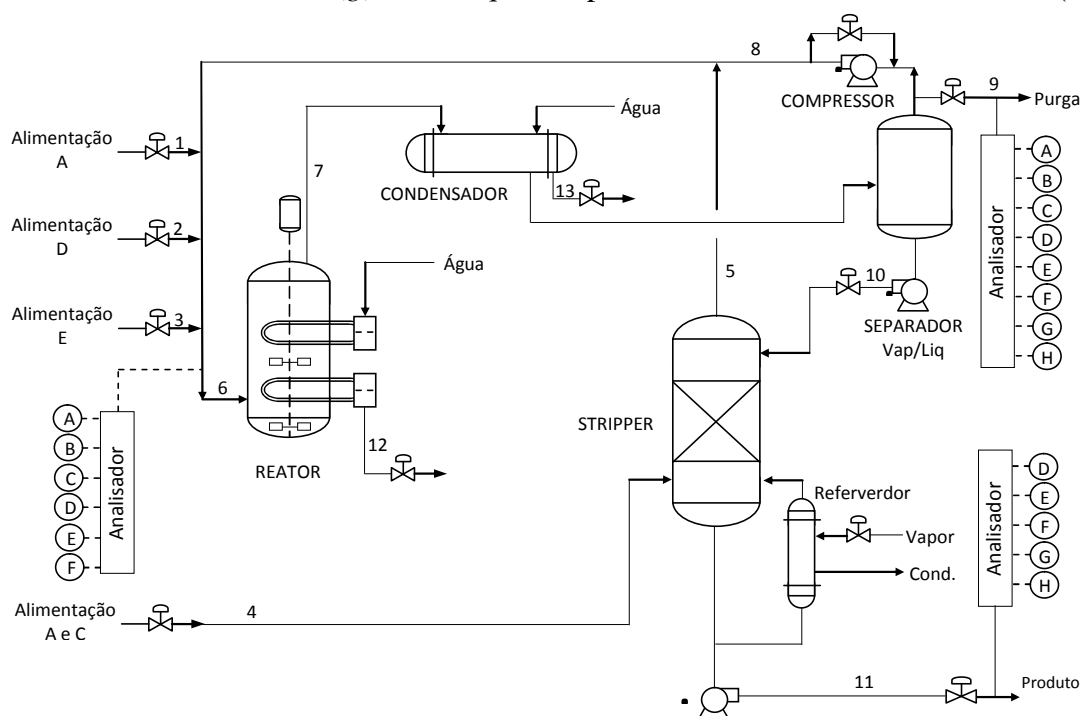
## 5.4 Estudos de Caso

Esta seção apresenta os resultados obtidos pelo sistema FDD-pRAM e compara o seu desempenho com o sistema FDD-WiSARD. Para isso são considerados dois estudos de caso, com dados simulados. Um deles, o *Tennessee Eastman Process* (TEP) compreende num *benchmark* na área de detecção e diagnóstico de falhas e o outro simula um tanque agitado

continuamente, referenciado como *Continuous Stirred Tank Reactor (CSTR)* e descritos nas seções 5.4.1 e 5.4.2, respectivamente.

### 5.4.1 O Processo Tennessee Eastman (TEP)

O TEP foi proposto originalmente por Downs e Vogel (1993) e possui características marcantes como a não linearidade de comportamento e a instabilidade em malha aberta. O modelo TEP possui uma forte interação entre as unidades que o constituem, com destaque para o “reator”, o “condensador de produto”, o “separador líquido/ vapor”, o “compressor de reciclo” e a “coluna de separação de produtos”, como mostrado na Figura 5.6. O objetivo do TEP é a simulação da produção de *G* e *H* a partir dos reagentes *A*, *C*, *D* e *E*, gerando ainda um subproduto *F*, de acordo com as reações mostrads nas Equações 5.16 a 5.19. Além dos reagentes e produtos é utilizado também um inerte gasoso *B*, introduzido nas correntes de alimentação dos reagentes. Para a simulação do processo químico com a planta TEP são utilizadas 12 variáveis manipuláveis e 41 variáveis medidas. Dentre as variáveis medidas, 22 são consideradas contínuas por apresentarem uma periodicidade de amostragem constante e de 3 minutos. Essas variáveis contínuas correspondem às medições de vazão, temperatura e pressão. As demais variáveis medidas têm períodos de amostragem maiores, e consistem dos analisadores de alimentação para o reator, a purga e o produto final.



**Figura 5.6:** Diagrama Esquemático do *Tennessee Eastman Process* (Oliveira *et al.*, 2017).

Como mostrado na Figura 5.6, o reator é alimentado por meio dos reagentes inseridos pelos fluxos 1, 2 e 3 e equipado com uma serpentina de resfriamento, onde circula água, e um agitador com velocidade variável. A corrente de saída do reator segue para o condensador pelo fluxo 7, onde ocorre uma liquefação parcial por resfriamento. A parte condensada é constituída,



principalmente, de produtos e uma fração pequena de reagentes. O condensador tem apenas uma corrente de saída, com duas fases, que segue para o separador líquido/ vapor. Parte da fração gasosa é purgada para evitar acumulação de inerte e subproduto no processo e o restante é enviado para o compressor centrífugo e reciclado para o reator. A parte líquida segue para a *stripper* (coluna retificadora) pelo fluxo 10, onde os reagentes ainda presentes na fase líquida são removidos com a alimentação de *A* e *C* e uma pequena quantidade de um inerte *B* (principal responsável pela purga no sistema) através do fluxo 4 que circula em contra-corrente com a fase líquida. Os reagentes removidos na *stripper* são enviados ao reator para reciclagem e a parte líquida, constituída basicamente por *G* e *H*, consiste no produto final do processo.

No trabalho original de Downs e Vogel (1993) são considerados seis modos de operação, diferenciados pelo caudal ou composição do produto, conforme apresentados na Tabela 5.1. Ricker (1995) usou uma planta em malha fechada para encontrar os estados estacionários para os seis modos de operação apresentados por Downs e Vogel (1993). Um desses estados estacionários, denominado por Ricker (1995) de “caso 1” e referente ao caso base, foi usado na simulação dos dados para o treinamento e validação do sistema FDD-pRAM proposto. O controle estacionário dos modos de operação tanto em Downs e Vogel (1993) quanto em Ricker (1995) considera 20 diferentes condições de falhas, conforme mostradas na Tabela 5.2. O sistema FDD-pRAM aqui proposto avalia apenas as 13 primeiras, por considerar que as demais falhas são mais complexas e resultantes de combinações de duas ou mais das 13 primeiras condições de falhas ou de mudanças nas variáveis (*setpoint*) de entrada.

**Tabela 5.1:** Modos de Operação para o *Tennessee Eastman Process* (TEP)

Modo de Operação	Razão Mássica G/H	Caudal do Produto G/H
1	50/50 (caso base)	7038kg/h / 7038kg/h
2	10/90	1408kg/h / 12669 kg/h
3	90/10	10000kg/h / 1111 kg/h
4	50/50	Produção Máxima
5	10/90	Produção Máxima
6	90/10	Produção Máxima

**Tabela 5.2:** Falhas Simuladas no TEP (Oliveira *et al.*, 2017)

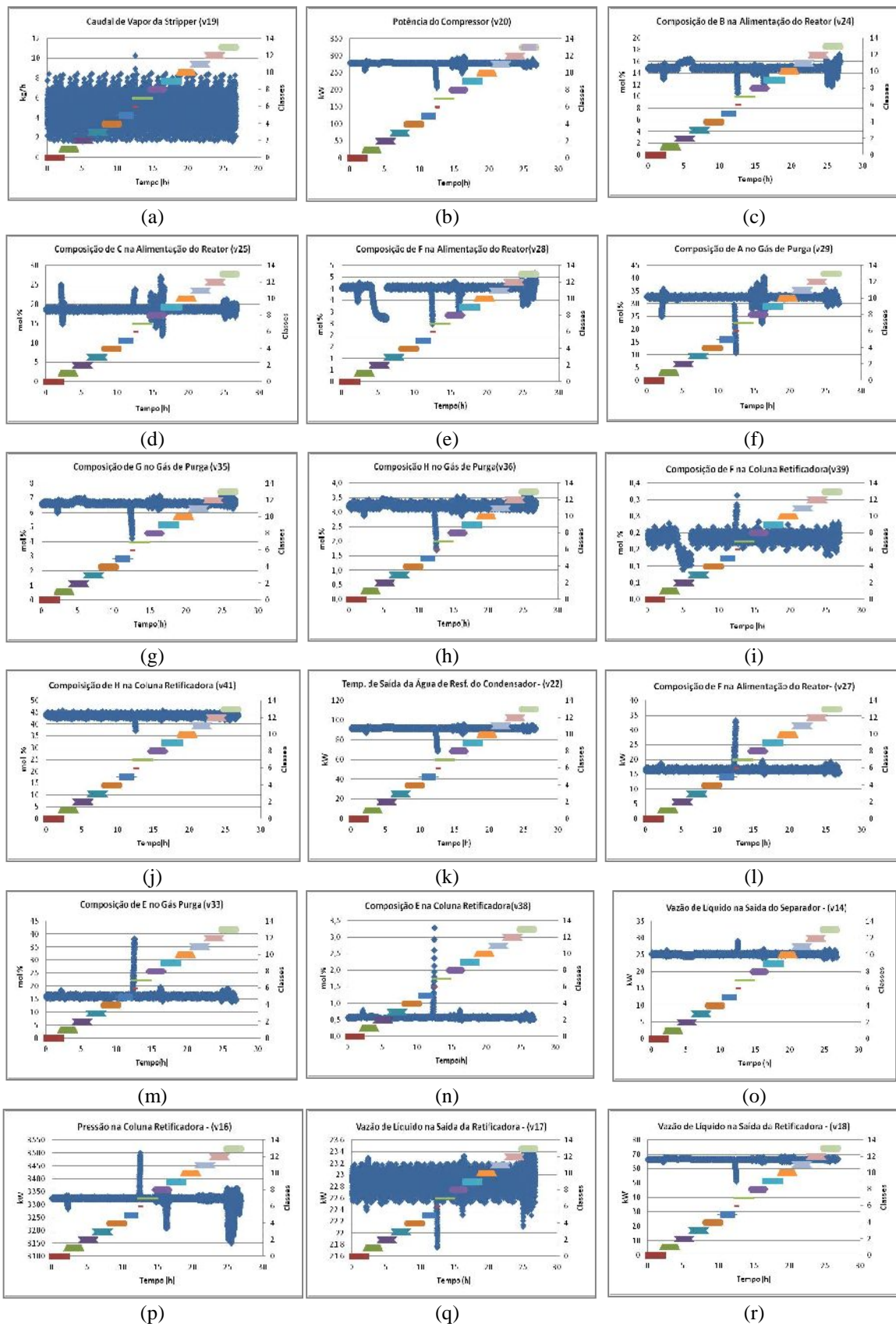
Falha	Descrição	Alteração
1	Razão <i>A/C</i> , mantendo <i>B</i> constante	Degrau
2	Composição de <i>B</i> , mantendo a razão <i>A/C</i> constante	Degrau
3	Temperatura de alimentação de <i>D</i>	Degrau
4	Temperatura da água de refrigeração para o reator	Degrau
5	Temperatura da água de refrigeração para o condensador	Degrau
6	Carga (válvula) de alimentação de <i>A</i>	Degrau
7	Carga (válvula) de alimentação de <i>C</i>	Oscilatória
8	Composições das alimentações de <i>A</i> , <i>B</i> e <i>C</i>	Oscilatória
9	Temperatura de alimentação de <i>A</i>	Oscilatória
10	Temperatura de alimentação de <i>C</i>	Oscilatória
11	Temperatura da água de refrigeração para o reator	Oscilatória
12	Temperatura da água de refrigeração para o condensador	Oscilatória
13	Variação cinética da reação (velocidade/componentes/ fatores)	Oscilatória

14	Válvula de água de arrefecimento do reator	Desconhecida
15	Válvula de água de refrigeração do condensador	Desconhecida
16	Desconhecida	Desconhecida
17	Desconhecida	Desconhecida
18	Desconhecida	Desconhecida
19	Desconhecida	Desconhecida
20	Desconhecida	Desconhecida

#### 5.4.1.1 Resultados e Análises - TEP

Os dados usados no treinamento e validação do sistema FDD-pRAM para o estudo de caso TEP foram obtidos com o simulador proposto por Ricker (1995). Os dados foram simulados com base no modo de operação “caso base”, proposto por Downs e Vogel (1993) com a estabilização apresentada por Ricker (1995). Para isso, foram gerados 5 (cinco) conjuntos de dados. Desses, um foi usado no treinamento e 4 na validação do sistema FDD-pRAM. O conjunto usado no treinamento foi gerado com os valores originais e os de validação foram gerados com variações de  $\pm 3\%$  e  $\pm 5\%$  na variável “taxa de produção”. A taxa de produção é considerada a variável controlada de maior relevância na composição dos produtos finais  $G$  e  $H$  (Ricker, 1995). Porém, variações fora do intervalo  $\pm 5\%$  tiram a planta do estado estacionário considerado. Com exceção da falha 6, e para cada conjunto de dados gerados, foram realizadas 14 simulações de 1000 períodos de amostragem de 3 minutos. Uma dessas simulações refere-se ao estado de operação normal e as outras aos 13 tipos de falhas considerados. Com o intuito de amenizar os efeitos de inicialização das simulações foram descartados os 20 primeiros períodos de cada uma delas (Oliveira *et al.*, 2017).

Os gráficos da Figura 5.7 ilustram parte do conjunto de dados usados no treinamento do sistema FDD-pRAM para o estudo de caso TEP. Com base na configuração de melhor desempenho do sistema FDD-pRAM, o algoritmo RecPun modificado selecionou 23 variáveis (atributos) para cada uma das 14 classes (modos de operação) consideradas na nessa aplicação. A Figura 5.7 ilustra o modo de operação “Normal” e um tipo de falha, mais precisamente a “Falha 2”, as demais classes (modos de falha) possuem comportamento semelhante. Das 23 variáveis selecionadas para o estado de operação Normal e das 23 para o estado de Falha 2, 19 são comuns. Das 19, foram escolhidas 10 de forma aleatória e representadas nas Figuras 5.7(a) a 5.7(j). Já as Figuras 5.7(k) a 5.7(n) são as demais variáveis relativas ao modo de operação Normal, e as Figuras 5.7(m) a 5.7(p) ilustram as outras variáveis selecionadas para o modo de operação Falha 2.



**Figura 5.7:** Dados Utilizados no Treinamento dos Sistemas FDD para o TEP: Modos de Operação “Normal” e “Falha 2”.

As Tabelas 5.3 e 5.4 apresentam, respectivamente, os percentuais de acerto nos diagnósticos e os parâmetros usados pelo sistema FDD-pRAM para a aplicação TEP. O percentual em cada

célula da Tabela 5.3 corresponde ao melhor desempenho para uma dada configuração (modelo de arquitetura *versus* variação na taxa de produção). Já os parâmetros apresentados na Tabela 5.4 estão relacionados com a configuração de melhor desempenho para cada arquitetura.

**Tabela 5.3:** Percentuais de Acerto para a Rede WiSARD pRAM para a Aplicação TEP (%)

Arquiteturas	Variação na Taxa de Produção				
	-5	-3	+3	+5	Média
1	98,63	<b>99,89</b>	99,25	95,25	98,17
2	95,95	<b>99,04</b>	97,01	90,37	97,11

**Tabela 5.4:** Parâmetros de Melhor Desempenho para o Sistema FDD-pRAM com Base nos Modelos de Arquitetura e na WiSARD com nodos pRAM para a Aplicação TEP

Arquiteturas	Parâmetros Usados						
	i	ii	iii	iv	v	vi	vii
1	23	12	273	3	40	1	-3
2	10	12	120	3	40	2	-3

i)  $n^\circ$ -variáveis(atributos)-selecionadas; ii) representação binária dos pontos  $m$  focais ( $n$ .bits); iii) padrão-entrada-rede ((i) x  $m = n$ .bits); iv) entradas-RAM ( $n$ . bits); v)  $n^\circ$ -RAM-por-discriminadores ((iii)÷(iv)); vi) *bleaching*-simples (1)/ percentual (2). vii) variação-taxa-de-produção (%).

Com base nos dados das Tabelas 5.3 e 5.4, são feitas as seguintes considerações:

a) As duas arquiteturas testadas apresentaram resultados satisfatórios para a aplicação TEP, com destaque para a arquitetura 1. A variação de -3% na taxa de produção apresentou maior consistência nas duas arquiteturas;

b) Nos sistemas propostos, o parâmetro *bleaching*, além de amenizar o desempate entre os discriminadores da rede, também permite determinar o nível de sensibilidade quando da comparação das probabilidades armazenadas nos nodos pRAM (ou seja, definir a saída do nodo pRAM). Assim, quanto maior a representação ( $n^\circ$  de dígitos) decimal dessas probabilidades, maior é a sensibilidade na definição das saídas dos nodos pRAM. Além disso, a adaptação da arquitetura 1 com o *bleaching* homogêneo está relacionada à mesma quantidade de padrões de treinamento para as classes (modos de operação) consideradas na aplicação TEP. Já a arquitetura 2 se adaptou melhor ao *bleaching* heterogêneo, pois na etapa de detecção de falhas (primeira camada da rede) a quantidade de padrões normais e de falhas difere significativamente, embora na etapa de diagnóstico de falhas (segunda camada da rede) as quantidades sejam equivalentes;

c) Com relação ao parâmetro taxa de produção, o desempenho do sistema de FDD-pRAM é inversamente proporcional ao percentual de variação na taxa de produção, o que justifica a recomendação dada por Ricker (1995) em não usar variações fora do intervalo de  $\pm 5\%$ .

As Tabelas 5.5 e 5.6 detalham os diagnósticos por classe obtidos com a RNSP WiSARD com nodos pRAM e dados simulados com variação de -3% na taxa de produção. Com finalidade comparativa, os percentuais apresentados nas Tabela 5.5 e 5.6 foram obtidos com o uso do algoritmo de seleção de atributos “RecPun modificado” e “RecPun original”, respectivamente.

**Tabela 5.5:** Matriz de Classificação para a RNSP WiSARD com nodos pRAM e Variação de -3% na Taxa de Produção com Algoritmo de Seleção de Atributos RecPun Modificado

Classes (Real)	Saídas Preditas pelo Sistema													Acertos (%)	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12		C13
C0	980														100,00
C1		980													100,00
C2			980												100,00
C3				978					1					1	99,80
C4					974			6							99,39
C5						975			5						99,49
C6							126								100,00
C7								979	1						99,90
C8									980						100,00
C9										980					100,00
C10											980				100,00
C11												980			100,00
C12													980		100,00
C13														980	100,00
Percentual de Acertos na Saída da Rede WiSARD com nodos pRAM (%)															<b>99,89</b>

**Tabela 5.6:** Matriz de Classificação para a RNSP WiSARD com nodos pRAM e Variação de -3% na Taxa de Produção com Algoritmo de Seleção de Atributos RecPun Original

Classes (Real)	Saídas Preditas pelo Sistema													Acertos (%)	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12		C13
C0	117			99	114	146		138		120		137	109		11,94
C1		980													100,00
C2			980												100,00
C3	117			121	135	113		100		107		136	151		12,35
C4	134			109	136	112		120		101		141	127		13,88
C5	122			101	134	97		145		119		132	130		9,90
C6							126								100,00
C7	118			75	99	123		133		98		200	134		13,57
C8									980						100,00
C9	116			84	120	133		151		111		135	130		11,33
C10											980				100,00
C11	59			67	70	75		82		100		351	176		35,82
C12	114			66	120	122		119		118		162	159		16,22
C13														980	100,00
Percentual de Acertos na Saída da Rede WiSARD com nodos pRAM (%)															<b>48,56</b>

Os percentuais apresentados nas Tabelas 5.5, quando comparados com os apresentados na Tabela 5.6, evidenciam a importância da modificação realizada no algoritmo RecPun para a detecção e o diagnóstico de falhas com o sistema FDD-pRAM. A confusão na classificação com o algoritmo RecPun original acontece porque todos, ou quase todos, os atributos selecionados para uma dada classe são também selecionados para outras classes. No sentido de fortalecer uma maior diferenciação nos atributos selecionados para as classes consideradas, como já mencionado na seção 5.3.2, uma restrição foi posta ao final do algoritmo original de modo que na relação biunívoca entre as classes possa existir no máximo 50% de atributos

comuns. Se a quantidade de atributos não for suficiente para uma substituição total ou parcial, a ordenação da segunda metade dos atributos comparados é modificada de forma aleatória.

A Tabela 5.7 compara os indicadores de qualidade obtidos com o sistema FDD-pRAM com os resultados apresentados por Oliveira *et al.* (2017) e Soares e Galvão (2010). Os resultados relativos ao sistema FDD-pRAM são os de melhor configuração e simulados com a variação de -3% na taxa de produção de *G* e *H*. Oliveira *et al.* (2017) usaram os mesmos dados de treinamento e validação com um sistema de FDD baseado na WiSARD com nodos RAM padrão e nodos com contadores como conteúdos de memória. Já Soares e Galvão (2010) realizaram a detecção e diagnóstico de falhas no TEP com o uso de outros dados e classificadores por grupos de falhas com características semelhantes. Os classificadores construídos por Soares e Galvão (2010) são baseados em métodos paramétricos (Análise Discriminante Linear, ADL, e Análise Discriminante Quadrática, ADQ) e não paramétricos (k Vizinhos mais Próximos, kNN, e Mínimos Quadrados Parciais, PLS). Na comparação, os percentuais de melhor desempenho estão em negritos.

**Tabela 5.7:** Indicadores de Qualidade na Validação do Sistema FDD-pRAM com a Rede WiSARD com Nodos pRAM e Variação de -3% na Taxa de Produção/Comparação com Soares e Galvão (2010)

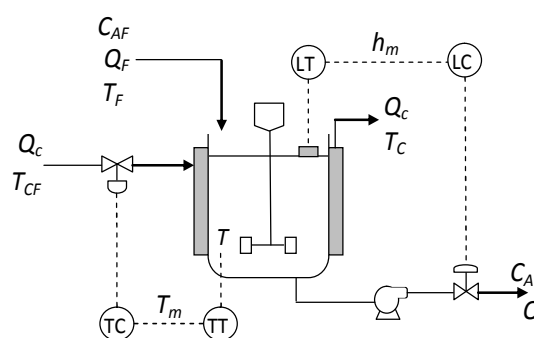
Itens Observados (pRAM)	FDD-pRAM		(Oliveira <i>et al.</i> 2017)		(Soares e Galvão, 2010)			
	RecPun Original	RecPun Modificado	RecPun Original	RecPun Modificado	Paramétricos		Não Paramétricos	
					APS-ADQ	APS-ADL	kNN	PLS
Padrões Testados	12866	12866	12856	12856	-	-		
Acertos - RNSP (n.)	6251	12752	5129	12788	-	-		
Acertos - RNSP (%)	48,58%	<b>99,89%</b>	39,90%	99,47%	-	-		
Filtro 1: Detecção de falhas (Média)	87,23%	<b>100 %</b>	99,42%	<b>100 %</b>	-	-		
• Estado de Operação Normal	11,94%	<b>100 %</b>	1,02%	<b>100 %</b>	-	-		
• Estado de Operação com Falha	93,44%	<b>100 %</b>	99,42%	<b>100 %</b>	-	-		
Filtro 2: Diagnóstico por Grupos de Falhas (Soares e Galvão, 2010)	55,37%	<b>99,90%</b>	58,02%	99,43%	93,20%	86,0%		
• Grupo 1 (falhas: 7, 8, 9 e 10)	62,58%	<b>100 %</b>	78,44%	99,87%	94,1%	85,0%	71,7 %	79,3 %
• Grupo 2 (falhas: 1 e 4)	56,94%	<b>99,69%</b>	51,84%	98,32%	97,8%	90,7%	72,2 %	77,3 %
• Grupo 3 (falhas: 2, 3, 5, 6 e 11)	56,82%	99,83%	43,21%	<b>99,85%</b>	94,9%	84,8%	74,6 %	80,8 %
• Grupo 4 (falhas: 12 e 13)	58,11%	<b>100 %</b>	53,83%	98,78%	86,0%	82,8%	70,6 %	75,9 %

Os dados apresentados na Tabela 5.7 confirmam o bom desempenho do sistema FDD-pRAM com resultados superiores aos obtidos pelo sistema de FDD proposto por Oliveira *et al.* (2017) e os diagnósticos por grupos de falhas apresentados por Soares e Galvão (2010).

Com os resultados e comparações apresentados nesta seção pode-se concluir que uma boa estrutura para o sistema FDD-pRAM para a aplicação TEP possui a seguinte configuração: “Seleção de Atributos” – uso do algoritmo RecPun modificado com 23 (vinte e três) atributos por classe; “Sistema de Inferência *Fuzzy*” com 23 (vinte e três) camadas de *clouds*; “Rede Neural sem Pesos” com uma única camada e estrutura baseada na rede WiSARD com 14 discriminadores (estado Normal e 13 tipos de falhas) com nodos pRAM, padrões de entrada para a rede igual a 273 *bits* ( $23 \cdot 12$ ), subpadrões de entradas para as RAM com 3 (três) *bits*, número de RAM por discriminadores igual a 92 ( $276/3$ ), uso do *bleaching* simples, e detecção do estado normal e de falha com diagnóstico por classe; “Filtro de Agrupamentos” com diagnóstico por grupos de falhas, de acordo com os definidos por Soares e Galvão (2010) e mostrados na Tabela 5.7. Os experimentos foram realizados numa única máquina básica da Intel (R), com processador *core* (TM) i5-2450M, CPU 2.50 GH, memória RAM de 4GB e sistema operacional de 64 *bits*. O tempo de execução do treinamento e teste do sistema para a referida configuração foi de 12 (doze) segundos. Os dados de treinamento e validação foram simulados com base no modo de operação “caso base” proposto por Downs e Vogel (1993) com a estabilização apresentada Ricker (1995) e variação de -3% na taxa de produção para o conjunto de validação.

#### 5.4.2 Reator Tanque Agitado Continuamente (CSTR)

O reator tanque agitado continuamente, referido como CSTR (*Continuous Stirred Tank Reactor*), é um tanque agitado com escoamento contínuo e sem acúmulo de reagentes ou produtos. Possui características do tipo: não isotérmico e com resfriamento jaqueta dinâmico; nível de líquido variável; composição de saída igual à do interior do reator; taxa da reação igual em todo o reator, inclusive na saída; uso preferencial em problemas com agitação intensa e reações em fase líquida. O modelo dinâmico do CSTR atende os princípios de mistura perfeita e de parâmetros físicos constantes e tem como base uma reação irreversível de primeira ordem clássica do tipo A → B. Um diagrama estrutural do CSTR e do sistema de controle é apresentado na Figura 5.8 (adaptado de Singhal e Seborg (2002)). Mais detalhes sobre a estrutura de controle e os parâmetros do controlador para o CSTR podem ser encontrados em Singhal e Seborg (2002) e Johannesmeyer (1999).



**Figura 5.8:** Diagrama Estrutural do Reator CSTR.

No contexto de detecção e diagnóstico de falhas, os reatores CSTR se configuram em um vasto campo de estudos com aplicações diversas. Como exemplo, o trabalho apresentado por Singhal e Seborg (2002) retrata uma base de dados típica para o estudo de caso CSTR, incluindo o estado de operação normal, 27 condições de falhas e 14 variáveis do processo. As 14 variáveis do processo e os 28 modos de operação considerados por Singhal e Seborg (2002) estão apresentados, respectivamente, nas Tabelas 5.8 e 5.9. As 4 últimas variáveis/ medições da Tabela 5.8 são sinais de saída do controlador e não são mostradas na Figura 5.8. O simulador usado neste trabalho tem como referência o trabalho de Singhal e Seborg (2002) e simula o

estado de operação normal e 17 estados dos 27 estados de falhas considerados. Os modos de operação aqui considerados estão em “negrito” na coluna 1 (um) da Tabela 5.9.

**Tabela 5.8:** Variáveis Medidas para o CSTR (Singhal e Seborg, 2002)

Variável	Descrição das Variáveis
$C_A$	Concentração da espécie A no reator (mol/L)
$T$	Temperatura do reator (K)
$T_c$	Temperatura do refrigerante na camisa de resfriamento (K)
$H$	Nível do líquido no reator (dm)
$Q$	Vazão de saída do reator (L/mim)
$Q_c$	Vazão do fluido refrigerante (L/mim)
$Q_F$	Vazão de alimentação do reator (L/mim)
$C_{AF}$	Concentração da espécie A na corrente de alimentação do reator (mol/L)
$T_F$	Temperatura de alimentação do reator (K)
$T_{CF}$	Temperatura de alimentação do fluido refrigerante (K)
$hC$	Medição do controlador de nível (mA); m = número de dados na janela atual
$QC$	Controlador de nível do reator (L/mim)
$TC$	Controlador de temperatura de saída (K)
$Q_cC$	Saída do controlador a partir do fluxo de resfriamento (mA)

**Tabela 5.9:** Modos de Operação para o CSTR (Singhal e Seborg, 2002)

Rótulos	Modos de Operação	Descrição	Valor Nominal
<b>N</b>	Operação normal	Operação nas condições normais, sem perturbações	N/A
<b>F1</b>	Desativação do catalisador	Rampa positiva na energia de ativação	A taxa da rampa para E/R é +3K/min
<b>F2</b>	Incrustação no trocador de calor	Rampa negativa na transferência de calor	A taxa da rampa para $U_{AC}$ é -125 J/(min·K)/min
<b>F3</b>	Medição do fluxo de líquido de resfriamento	A medição da vazão de resfriamento permanece com seu último valor	N/A
<b>F4 e F-4</b>	<i>Bias</i> na medição de temperatura do reator	A medição de temperatura do reator tem um <i>bias</i>	±4 K
<b>F5 e F-5</b>	<i>Stiction</i> (atrito estático) da válvula de resfriamento + <b>F7</b>	Banda morta para <i>stiction</i> = 5% do <i>span</i> da válvula	N/A
<b>F6 e F-6</b>	Mudança degrau em $Q_F$	Mudança degrau na vazão de alimentação	±10 L/min
<b>F7 e F-7</b>	Mudança rampa em $C_{AF}$	Rampas na concentração de alimentação para cima ou para baixo	A taxa da rampa é ±6·10 <sup>-4</sup> (mol/L)/min



<b>F8 e F-8</b>	Mudança rampa em $T_F$	Rampas na temperatura de alimentação para cima ou para baixo	A taxa da rampa é $\pm 0,1$ K/min
<b>F9 e F-9</b>	Mudança rampa em $T_{CF}$	Rampas na temperatura de alimentação de resfriamento para cima ou para baixo	A taxa da rampa é $\pm 0,1$ K/min
F10 e F-10	Mudança degrau em $P_{CU}$	Mudança degrau da pressão na linha de refrigeração	$\pm 2,5$ psi
F11 e F-11	Mudança degrau em $P_D$	Mudança degrau da pressão na linha de saída do reator	$\pm 5$ psi
<b>F12</b>	Oscilações amortecidas na vazão de alimentação	Mudanças na vazão de alimentação como $e^{-t/33} \cdot \text{sen}(2 \cdot \pi/10) \cdot \text{L/min}$	10 L/min
F13	Distúrbio auto regressivo na vazão de alimentação	$Q_F(k) = 0,8 \cdot Q_F(k-1) + w(k);$ $w(k) \sim N(0,1)^*$	N/A
<b>S1 e S-1</b>	Alteração do valor nominal de $T$	Alteração do valor nominal da temperatura do reator	$\pm 3$ K
<b>O1</b>	Oscilações de alta frequência na vazão de alimentação	Oscilações sustentadas com frequência de 3 ciclos/min	10 L/min
<b>O2</b>	Oscilações de frequência intermédia na vazão de alimentação	Oscilações sustentadas com frequência de 1 ciclo/min	10 L/min
<b>O3</b>	Oscilações de frequência intermédia na vazão de alimentação	Oscilações sustentadas com frequência de 0,5 ciclos/min	10 L/min
<b>O4</b>	Oscilações de baixa frequência na vazão de alimentação	Oscilações sustentadas com frequência de 0,2 ciclos/min	10 L/min

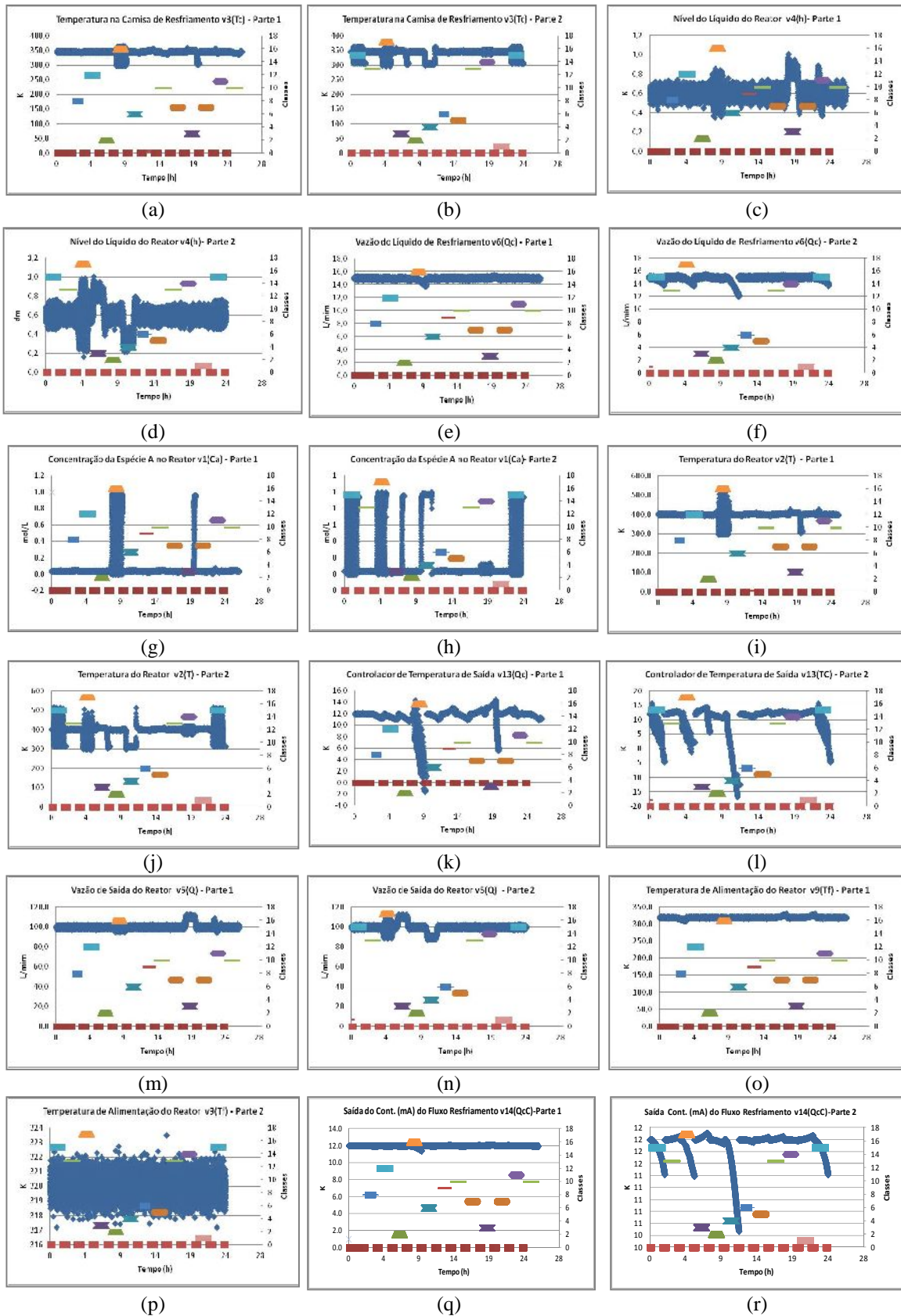
\* Curva normal com média 0 e desvio padrão 1.

#### 5.4.2.1 Resultados e Análises - CSTR

Para cada estado de operação (normal ou falha) foram simulados dados por um período de 120 minutos com amostragem de 5 segundos. A simulação dos 120 minutos para os estados com falha acontece na sequência normal/falha/ normal com os intervalos 15/ 85/ 20 minutos. Assim, um modo de operação com falhas só acontece com a certeza de uma inicialização e retomada do sistema para o estado de operação normal. A composição final dos conjuntos de treinamento e de validação contém 1 (um) período de 120 minutos (1440 pontos) para o estado de operação normal e 1 (um) ou 2 (dois) períodos, também de 120 minutos, para cada estado de falha. A decisão por considerar um ou dois períodos para cada estado de falha e o ordenamento temporal dos diversos períodos de 120 minutos considerados foram definidos aleatoriamente.

Para uma maior compreensão do comportamento dos dados de treinamento do sistema FDD-pRAM, os gráficos da Figura 5.9 ilustram os atributos selecionados com o algoritmo RecPun modificado para os modos de operação “Normal” e “Falha 2”. As variáveis apresentadas são relativas à configuração de melhor desempenho do sistema FDD-pRAM para a aplicação CSTR. Como o volume de dados de treinamento é muito grande, cada variável foi representada em duas séries de dados, aqui denominadas de Parte 1 e Parte 2. Assim, as Figuras 5.9(a) a 5.9(f) são variáveis comuns aos modos de operação Normal e Falha 2, as Figuras 5.9(g) a 5.9(k)

são as demais variáveis relativas ao modo de operação Normal e as Figuras 5.9(l) a 5.9(s) complementam as variáveis selecionadas para o modo de operação Falha 2.



**Figura 5.9:** Dados Utilizados no Treinamento dos Sistemas FDD para o CSTR: Modos de Operação “Normal” e “Falha 2”.

A Tabela 5.10 mostra os resultados obtidos com o sistema FDD-pRAM. Os valores ou percentuais apresentados para cada arquitetura correspondem à configuração de melhor desempenho do sistema, com base nos parâmetros descritos na seção 5.3.3. Os valores ou percentuais relativos ao diagnóstico por classes são saídas da rede pRAM e os relativos à detecção e grupos de falhas são obtidos como saída do sistema como um todo (rede mais filtro de agrupamento).

**Tabela 5.10:** Percentuais de Acerto do Sistema FDD-pRAM para a Aplicação CSTR (%)

Parâmetros	Arquitetura 1	Arquitetura 2	Média
Padrões Testados	34560	34560	34560
Acertos (n°.)	34518	32544	33531
Acertos (%)	<b>99,88</b>	<b>94,17</b>	97,03
<b>Deteção de Falhas</b>	<b>100,00</b>	99,91	99,96
Estado Normal	<b>100,00</b>	99,73	99,87
Estado de Falha	<b>100,00</b>	99,99	100,00
<b>Diagnóstico de Falhas</b>	<b>100,00</b>	99,73	99,87
F1	<b>100,00</b>	93,14	96,57
F2	<b>100,00</b>	53,63	76,82
F3	<b>100,00</b>	<b>100,00</b>	100,00
F4	<b>100,00</b>	<b>100,00</b>	100,00
F5	97,94	98,92	98,43
F6	<b>100,00</b>	99,90	99,95
F7	<b>100,00</b>	90,69	95,35
F8	<b>100,00</b>	<b>100,00</b>	100,00
F9	<b>100,00</b>	94,31	97,16
F10	<b>100,00</b>	83,82	91,91
F11	<b>100,00</b>	90,10	95,05
F12	<b>100,00</b>	93,82	96,91
F13	<b>100,00</b>	35,49	67,75
F14	<b>100,00</b>	99,80	99,90
F15	<b>100,00</b>	99,02	99,51
F16	<b>100,00</b>	<b>100,00</b>	100,00
F17	<b>100,00</b>	99,31	99,66
Tempo de Execução (s)	36	28	35

A Tabela 5.11 contém os parâmetros usados para a configuração de melhor desempenho do sistema FDD-pRAM para os dois modelos de arquiteturas.

**Tabela 5.11:** Parâmetros de Melhor para a Configuração do Sistema FDD-pRAM com Base nos Modelos de Arquitetura para a Rede WiSARD pRAM com a Aplicação CSTR

Arquiteturas	Parâmetros Avaliados						
	i	ii	iii	iv	v	vi	vii
1	6	12	72	12	6	2	-
2	6	12	72	12	6	2	-

i)  $n^{\circ}$ -variáveis(atributos)-selecionadas; ii) representação binária dos pontos  $m$  focais ( $n^{\circ}.bits$ ); iii) padrão-entrada-rede ((i) x  $m = n^{\circ}.bits$ ); iv) entradas-RAM ( $n^{\circ}.bits$ ); v)  $n^{\circ}$ -RAM-por-discriminadores ((iii)÷(iv)); vi) *bleaching*-simples (1)/percentual (2). vii) variação-taxa-de-produção (%).

Com os resultados apresentados nas Tabelas 5.10 e 5.11 é possível fazer as seguintes considerações:

- a) O sistema FDD-pRAM com a arquitetura 1 (um) apresentou melhor desempenho para a aplicação CSTR, embora as duas arquiteturas tenham se adaptado bem à detecção e ao diagnóstico de falhas para o referido estudo de caso. O percentual de 99,88% no reconhecimento e classificação dos padrões testados demonstra a boa adaptação das RNSP e do sistema como um todo para o estudo de caso CSTR;
- b) O item vi da Tabela 5.11 aponta o *bleaching* heterogêneo como o de melhor desempenho nas duas arquiteturas. Esse comportamento é esperado, pois a quantidade de padrões treinados em cada discriminador difere em duas situações. A primeira está relacionada à quantidade de padrões com o modo de operação normal formada por um ciclo de simulação mais os 15 e 20 minutos de simulação inicial e final para cada modo de operação com falha. A segunda diz respeito ao uso de um ou dois ciclos de padrões, usados de forma aleatória, para os modos de operação com falha. Além disso, a primeira camada da arquitetura 2 já diferencia naturalmente a quantidade de padrões de treinos em normal e de falha;
- c) Outro comportamento também esperado é a conectividade baixa dos nodos pRAM (2 a 5 entradas, aproximadamente). Porém, tanto na aplicação TEP quanto na CSTR os nodos pRAM desempenharam melhor com 12 *bits* de entrada, o que equivale a 4096 endereços de memória. Uma conectividade baixa evita um espalhamento dos endereços acessados. No entanto, esse espalhamento elevado é devido ao grande volume de dados de treinamento nas duas aplicações.
- d) Também, os incrementos nos conteúdos de memória com uso das probabilidades impedem o problema da saturação enfrentado pela maioria dos nodos RAM-based.

As Tabelas 5.12 e 5.13 apresentam, respectivamente, os diagnósticos por classe obtidos na saída da rede WiSARD com nodos pRAM e uso dos algoritmos de seleção de atributos “RecPun modificado” e “RecPun original” (taxa de acertos na classificação de padrões em negrito nas Tabelas 5.11 e 5.12).

**Tabela 5.12:** Matriz de Classificação para a Rede WiSARD com nodo pRAM com o Algoritmo de Seleção de Atributos RecPun Modificado para a Aplicação CSTR

Classe (Real)	Saídas Preditas com a RNSP-pRAM																	Acertos (%)	
	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16		F17
F0	11100																		99,86
F1		1020																	99,51
F2			1020																99,51
F3				1020															100,00
F4					2040														99,75
F5						1998						42							99,41
F6							2040												99,71
F7								2040											99,75
F8									1020										99,80
F9										1020									99,90
F10											2040								99,66
F11												1020							99,31
F12													1020						99,80
F13														1020					98,92
F14															1020				99,90
F15																1020			99,12
F16																	2040		99,95
F17																		1020	100,00
<b>Percentual de Acertos da Rede</b>																			<b>99,88</b>

**Tabela 5.13:** Matriz de Classificação para a Rede WiSARD com nodo pRAM com o Algoritmo de Seleção de Atributos RecPun Original para a Aplicação CSTR

Classe (Real)	Saídas Preditas com a RNSP-pRAM																		
	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	Acertos (%)
F0	11070															8	7	15	99,73
F1		950		21										49					93,14
F2			547										473						53,63
F3				1020															100,00
F4					2040														100,00
F5				4		2018			8		2				8				98,92
F6							2038											2	99,90
F7								1850		190									90,69
F8									1020										100,00
F9									56		962							2	94,31
F10		4								326		1710							83,82
F11									50		51		919						90,10
F12			63											957					93,82
F13		630		22						4	1	1			362				35,49
F14																1018		2	99,80
F15	1															1010	8	1	99,02
F16																	2040		100,00
F17																1	6	1013	99,31
<b>Percentual de Acertos da Rede</b>																			<b>94,17</b>

Dos 27 estados de falhas abordados por Singhal e Seborg (2002) somente 17 foram considerados no simulador CSTR desenvolvido e utilizado neste trabalho. O referido simulador encontra-se em processo de construção com o objetivo de completar a simulação com os estados de falhas ainda não simulados. O método proposto por Singhal e Seborg (2002) objetiva a correspondência de padrões em uma base de dados de séries temporais multivariada. O modelo proposto localiza períodos de dados semelhantes quando comparados a um período atual arbitrariamente selecionado. Para isso, utiliza-se de um fator de similaridade PCA, associado a uma métrica proposta pelos autores para determinar de forma não supervisionada a distância entre dois conjuntos de dados. Para a validação da metodologia proposta os dados são processados em fluxos sem a necessidade de treinamento do sistema. Diferentes métodos de correspondência de padrões são comparados em termos de precisão (número de registros corretamente identificados) e de eficiência (capacidade de localização de registros semelhantes na base de dados histórica). Para a metodologia proposta, a precisão e eficiência são de 69,00% e 79,00%, respectivamente, com valor médio de 74% para os 28 modos operacionais considerados no estudo de caso. Em média o desempenho do método proposto foi superior quando comparado com outros métodos estatísticos multivariados testados pelos autores. Por outro lado, a despeito de terem propósitos distintos, a precisão na detecção e diagnóstico de falhas com o sistema FDD-pRAM foi de 98,88%, bastante superior à precisão apresentada por Singhal e Seborg (2002).

Os resultados apresentados pelo sistema FDD-pRAM com o algoritmo de seleção de atributos “RecPun modificado” foram superiores em comparação com o uso do algoritmo “RecPun original”. A confusão ocorrida com o algoritmo “RecPun original” tem o inconveniente de selecionar os mesmos atributos para duas ou mais classes, dentre as consideradas. Como já mencionado na seção 5.3.2 e no estudo de caso TEP, a modificação realizada ao final do algoritmo RecPun original evita ou ameniza essa inconveniência e contribui significativamente para o bom desempenho do sistema FDD-pRAM, tanto para a aplicação CSTR quanto para o TEP.

Com os resultados e as considerações apresentadas, o sistema FDD-pRAM proposto para o estudo de caso CSTR possui a seguinte configuração: “Seleção de atributos” por classe com o algoritmo RecPun modificado; “Sistema de inferência *fuzzy*” com 6 camadas de *clouds* geradas a partir dos atributos selecionados; “Rede neural sem pesos” com a arquitetura 1 (um) e modelo WiSARD com nodos pRAM, padrões de entrada para a rede igual a 72 *bits* (12·6), subpadrões de entradas para as RAM com 3 (três) *bits*, número de RAM por discriminadores igual a 24 (72/3), uso do *bleaching* heterogêneo, diagnóstico por classe nas saídas da rede; “Filtro de agrupamento” como etapa final do sistema usado apenas para a classificação de um padrão de entrada como estado de operação normal ou estado de falha, pois nenhum agrupamento de falhas foi considerado para o modelo CSTR. O tempo de execução do treinamento e teste com sistema para essa configuração foi de 22 segundos, utilizando a mesma configuração computacional utilizada na aplicação TEP.

## 5.5 Conclusões Parciais

Este trabalho abordou o problema de detecção e diagnóstico de falhas em sistemas dinâmicos multivariáveis. Para isso, foi proposto um sistema, denominado de FDD-pRAM, composto por “classificadores neurais sem pesos com nodos probabilísticos” e por um “pré-processamento de dados” baseado na “seleção de atributos” e na “inferência *fuzzy*” para a granulação dos atributos selecionados. O sistema FDD-pRAM foi validado com dados simulados para os estudos de caso TEP e CSTR. Com os resultados apresentados é possível fazer as seguintes

considerações: a seleção de atributos por classe e o sistema de inferência *fuzzy* usados no pré-processamento dos dados contribuíram significativamente para a formação e diferenciação dos “padrões comportamentais” usados no treinamento e validação da RNSP probabilística; o fato de um dado discriminador na rede WiSARD aprender somente os padrões/ atributos selecionados para uma mesma classe fortalece a diferenciação dos padrões de entrada tanto no treinamento quanto na validação da rede; as probabilidades contínuas armazenadas nos nodos pRAM dos discriminadores lidam melhor com os ruídos presentes nos dados e aumenta o nível de sensibilidade no processo de reconhecimento e classificação de padrões pelos discriminadores; a técnica de *bleaching* com os nodos pRAM ameniza o uso da aleatoriedade entre os discriminadores da rede WiSARD com um nível de sensibilidade maior quando comparado a outros nodos RAM-based.

Além disso, nos dois estudos de caso, a arquitetura 1 (um) foi a de melhor desempenho para a detecção e o diagnóstico de falhas. Essa arquitetura reconhece/ discrimina o padrão de teste como de operação normal ou de uma falha específica. Assim, a detecção (reconhecimento) e o diagnóstico (classificação) do padrão de teste são realizados simultaneamente. Esse procedimento simultâneo difere da maioria dos modelos de FDD, em que o diagnóstico é realizado numa etapa posterior à detecção, como na arquitetura 2 usada neste trabalho. Diferentemente dos modelos neurais com pesos, o treinamento e teste na rede WiSARD, e por consequência no sistema como um todo, são realizados em um único procedimento (única apresentação), conferido assim consistência aos resultados obtidos.

O sistema FDD-pRAM proposto realiza a detecção e o diagnóstico em sistemas dinâmicos multivariados e foi validado com os estudos de caso considerados. No entanto, para cada aplicação faz-se necessária a configuração do sistema FDD-pRAM de modo a atender especificidades tais como: dimensionalidade; estados operacionais (classes ou grupos) considerados; dados de treinamento e validação; entre outros.

Com essas considerações finais conclui-se que as redes neurais sem pesos, com a contribuição da seleção de atributos e do sistema de inferência *fuzzy* para a granulação dos atributos selecionados, adaptou-se bem para o problema de detecção e diagnóstico de falhas em sistemas dinâmicos multivariáveis, o que foi comprovado com as taxas de acerto na classificação de 99,89% e 99,88% pelo sistema FDD-pRAM para as aplicações TEP e CSTR, respectivamente.

## Abreviaturas

ADL:	Análise Discriminante Linear.
ADQ:	Análise Discriminante Quadrática.
AG:	Algoritmo Genético.
APS:	Algoritmo de Projeções Sucessivas.
arg-max:	Argumento máximo.
BA:	Estado da Bahia – Brasil.
CAPES:	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.
CNPq:	Conselho Nacional de Desenvolvimento Científico e Tecnológico.
CPU:	<i>Central Processing Unit</i> .
$Disc_k$ :	Discriminador $k$ .
FDD:	Fault Detection and Diagnosis.
FDD-pRAM:	Sistema de Detecção e Diagnóstico de Falhas com Rede Neural Sem Peso Granular.



FDI:	<i>Fault Detection and Isolation.</i>
GB:	GigaByte.
i-pRAM:	<i>integrating pRAM.</i>
kNN:	k-Nearest Neighbour.
MCP:	<i>McCulloch-Pitts.</i>
MPLN:	Multiple-valued Probabilistic Logic Neuron.
pRAM:	<i>probabilistic RAM.</i>
$Pun_i$ :	Punição para o atributo $i$ .
QA:	Quantidade de atributos(variáveis) considerados num estudo de caso.
QC:	Número de classes consideradas num estudo de caso.
RAM:	<i>Random Access Memories.</i>
Rec:	Recompensa para o atributo $i$ .
RecPun:	RECompensa/ PUNição.
RNA:	Rede Neural Artificial.
RNSP:	Rede Neural Sem Pesos.
RP:	Recompensa/Punição.
TEP:	<i>Tennessee Eastman Process.</i>
UFBA:	Universidade Federal da Bahia.
WiSARD:	<i>Wilkes-Stonham-Aleksander Recognition Device.</i>

## Notações

$a^h$ :	Atributo da série original no período $h$ .
$x_{jm}^t$ :	Atributo $m$ para o padrão de entrada $x$ no tempo $t$ e função de pertinência $j$ .
$x_{jm}^{t+\Delta}$ :	Atributo $m$ para o padrão de entrada $x$ no tempo $t+$ e função de pertinência $j$ .
$\bar{a}_i$ :	Atributo médio dos $t$ valores da série processada para o atributo $i$ .
$a_n^t$ :	Atributo $n$ do padrão de entrada $x^t$ .
$A_{jm}^i$ :	Função de pertinência $j$ para o atributo $m$ no grânulo $i$ .
$l_{jm}^i$ :	Limite inferior para o atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$L_{jm}^i$ :	Limite superior para o atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$x^t$ :	Padrão de entrada $x$ no instante $t$ .
$2^p$ :	Quantidade de memória na RAM locadas por $p$ bits de entradas.
A(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
$a$ :	Endereço formado pelo padrão binário I.
B(g):	Inerte gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
C(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
C[ $a$ ] :	Conteúdo de memória acessado pelo vetor de entrada $a$ .
$C_k$ :	Classe relacionada ao discriminador $k$ .
D(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
E(g):	Reagente gasoso utilizado no simulador <i>Tennessee Eastman Process</i> .
F(g):	Subproduto gasoso gerado no simulador <i>Tennessee Eastman Process</i> .

$f(x)$ :	Função que define aproximadamente os limites para as classes consideradas.
$F_j$ :	Padrão de Falha do Tipo $j$ para a aplicação utilizada.
$g(r)$ :	Valores desejados como saídas para o nodo MPLN.
$G$ :	Produto de saída no simulador <i>Tennessee Eastman Process</i> .
$h$ :	Horizonte da janela temporal processada.
$H$ :	Produto de saída no simulador <i>Tennessee Eastman Process</i> .
$I$ :	Vetor de entradas para as RAM.
$k$ :	Quantidade de discriminadores na rede WiSARD.
kg/h:	Quilograma por hora.
kPa:	Unidade de medida internacional Quilo Pascal.
kscmh:	Mil unidades de metros cúbicos por hora.
kW:	Quilowatt é uma unidade de potência correspondente a 10000 watts.
$m$ :	Número de atributos selecionados pelo algoritmo RecPun.
$m^3/h$ :	Metro cúbico por hora.
mol%:	Unidade utilizada para a grandeza quantidade de substância ou matéria.
$n$ :	Dimensionalidade dos vetores de entrada para os estudos de caso considerados.
$N$ :	Estado de operação normal para os estudos de caso considerados.
$p$ :	Número de <i>bits</i> do vetor de entradas para as RAM.
$P_{2,max}$ :	Segunda maior pontuação obtida por um dos discriminadores da rede WiSARD.
$P_{max}$ :	Pontuação máxima obtida por um dos discriminadores da rede WiSARD.
$r$ :	Numero de RAM em um discriminador.
$R_{i,c}$ :	Posição do atributo $i$ na classe $c$ .
$RP_i$ :	Recompensa e Punição para o atributo $i$ .
$V_{i,c}$ :	Valor que representa a importância do atributo $i$ para a classe $c$ .
$x$ :	Padrão de entrada para o sistema.
$y$ :	Saída da rede neural sem pesos com valor igual a um rótulo de classe.

## Símbolos Gregos

$\gamma$ :	A taxa de aprendizado para o nodo MPLN.
$\gamma_m^i$ :	Grânulo $i$ para o atributo $m$ .
$\theta_{a_m}^i$ :	Grau de compatibilidade do atributo $m$ no grânulo $i$ .
$\theta^i$ :	Grau de compatibilidade do grânulo $i$ .
$\theta^t$ :	Grau de compatibilidade do padrão de entrada $x$ no tempo $t$ .
$\mu(x_{jm}^i)$ :	Grau de pertinência do atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$\Lambda$ :	Operador da função t-norma produto.
$\beta^t$ :	Padrão comportamental do padrão de entrada $x$ no tempo $t$ .
$\beta_{a_m}^i$ :	Ponto focal do atributo $m$ no grânulo $i$ .
$\beta^i$ :	Ponto focal para o grânulo $i$ .
$\lambda_{jm}^i$ :	Valor intermediário do atributo $m$ no grânulo $i$ para a função de pertinência $j$ .
$\Gamma_{jm}^i$ :	Valor intermediário do atributo $m$ no grânulo $i$ para a função de pertinência $j$ .

- $\Delta$  : Variação com valor inteiro positivo.
- $i$ : Valores possíveis de uma variável.
- : Parâmetro que define a precisão dos grânulos criados.
- : Operador de somatório de valores definidos.

## Referências

- Aleksander, I. (1966). Self-adaptive universal logic circuits. *Electronics Letters*, 2, 231.
- Aleksander, I. (1967). Adaptive systems of logic networks and binary memories. *Proceedings of the Spring Joint Computer Conference, ACM*, 30, 707-712.
- Aleksander, I. (1989). Canonical neural nets based on logic nodes. *In Proc. of IEE International Conference on Artificial Neural Networks*, 313, 110-114.
- Aleksander, I., Morton, H. (1990). An introduction to neural computing. *Chapman and Hall*, London.
- Aleksander, I., Thomas, W. V., Bowden, P. A. (1984). WISARD: a radical step forward in image recognition. *Sensor Review*, 4(3), 120-124.
- Angelov, P., Yager, R. (2011). A new type of simplified fuzzy rule-based system, *International Journal of General Systems*, 41(2), 163-185.
- Angelov, P., Zhou, X. (2006). Evolving fuzzy systems from data streams in real-time. *International Symposium on Evolving Fuzzy Systems*, 29-35.
- Blázquez, L. F., Miguel, L. J. (2005). Additive fault detection in nonlinear dynamic systems with saturation. *ISA Transactions*, 44(4), 515-538.
- Bledsoe, W. W., Browning, I. (1959). Pattern recognition and reading by machine. *Proceedings Eastern Joint Computer Conference*, 225-232.
- Burns, C. M. (2006). Towards proactive monitoring in the petrochemical industry. *Safety Science*, 44(1), 27-36.
- Cardoso, D. O., Carvalho, D. S., Alves, D. S., Souza, D. F., Carneiro, H. C., Pedreira, C. E., França, F. M. (2016). Financial credit analysis via a clustering weightless neural classifier. *Neurocomputing*, 183, 70-78.
- Carneiro, H. C., França, F. M., Lima, P. M. (2015). Multilingual part-of-speech tagging with weightless neural networks. *Neural Networks*, 66, 11-21.
- Chiang, L. H., Russel, E. L., Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. 1. Edição, Springer.
- Clarkson, T. G., Guan, Y., Taylor, J. G. (1993). Generalization in probabilistic RAM nets. *IEEE Transactions on Computers*, 4 (2), 360-363.
- Clarkson, T. G., Ng., C. K., Gorse, D., Taylor, J. G. (1992). Learning probabilistic RAM nets using VLSI structures. *IEEE Transactions on Computers*, 41(12), 1552-1561.
- Downs, J. J., Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3), 245-255.
- Fan, C. F., Yih, S., Tseng, W. H., Chen, W. C. (2013). Empirical analysis of software-induced failure events in the nuclear industry. *Safety Science*, 57, 118-128.

- Fernandez, A., Lopez, V., del Jesus, M. J., Herrera, F. (2015). Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*, 80, 109-121.
- Gorse, D., Romano-Critchley, D. A., Taylor, J.G. (1997). A pulse-based reinforcement algorithm for learning continuous functions. *Neurocomputing* 14(4), 319-344.
- Gorse, D., Taylor, J. G. (1988). On the equivalence and properties of noisy neural networks and probabilistic RAM nets. *Physics Letters*, 131(6), 326-332.
- Gorse, D., Taylor, J. G. (1991). A continuous input RAM-based stochastic neural model. *Neural Networks*, 4, 657-665.
- Hecht-Nielsen, R. (1990). Neurocomputing. USA: Addison-Wesley Publishing Company.
- Hong, T., Wang, P. (2014). Fuzzy interaction regression for short term load forecasting. *Fuzzy Optimization and Decision Making*, 13(1), 91-103.
- Isermann, R., Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5), 709–719.
- Johannesmeyer, M. C (1999). Abnormal situation analysis using pattern recognition techniques and historical data. *M.Sc. Thesis, University of California, Santa Barbara, CA*.
- Kan, W. K., Aleksander, I. (1987). A probabilistic logic neuron network for associative learning. *In Proc. of the IEEE International Conference on Neural Networks* 2, 541-548.
- Kar, S., Das, S., Ghosh, P. K. (2014). Applications of neuro fuzzy systems: A brief review and future outline. *Applied Soft Computing*, 15, 243-259.
- Kuncheva, L. I., Whitaker, C. J. (2003). Examining the relationship between majority vote accuracy and diversity in bagging and boosting. *Technical Report School of Informatics, University of Wales, Bangor*.
- Lee, S., Lee, E. (1975). Fuzzy neural networks, *Mathematical Biosciences*, 23(1-2), 151-177.
- Leite, D., Costa, P., Gomide, F. (2009). Evolving granular classification neural networks. *International Joint Conference on Neural Networks*, 1736-1743.
- Leite, D., Costa, P., Gomide, F. (2012). Evolving granular neural network for fuzzy time series forecasting. *The International Joint Conference on Neural Networks*, 1-8.
- Liu, H. T. (2007). An improved fuzzy time series forecasting method using trapezoidal fuzzy numbers. *Fuzzy Optimization and Decision Making*, 6(1), 63-80.
- Ludermir, T. B., Carvalho, A. C. P. L. F., Braga, A. P., de Souto, M. C. P. (1999). Weightless neural models: A review of current and past works. *Neural Computing Surveys*, 2, 41-61.
- McCulloch, W. S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115-137.
- Myers, C., Aleksander, I. (1988). Learning algorithms for probabilistic logic nodes. *In Abstracts of I Annual INNS Meeting*, p. 205.
- Myers, C., Aleksander, I. (1989). Output functions for probabilistic logic nodes. *In Proceedings of the IEE International Conference Neural Networks*, 310-314.
- Oliveira, J. C. M., Pontes, K. V., Sartori, I., Embiruçu, M. (2017). Fault Detection and Diagnosis in dynamic systems using Weightless Neural Networks. *Expert Systems with Applications*, 84, 200-219.

- Pedrycz, W., Gomide, F. (2007). Fuzzy systems engineering: Toward Human-Centric Computing. *Wiley-IEEE Press, 1<sup>a</sup> Edition*, 526 p.
- Ricker, N. L. (1995). Optimal steady-state operation of the Tennessee Eastman challenge process. *Computers and Chemical Engineering*, 19(9), 949-959.
- Romano, D., Kinnaert, M. (2006). Robust design of fault detection and isolation systems. *Quality and Reliability Engineering International*, 22(5), 527-538.
- Rosa, R., Gomide, F., Ballini, R. (2013). Evolving hybrid neural fuzzy network for system modeling and time series forecasting. *12th International Conference in Machine Learning and Applications (ICMLA)*, 2, 378-383.
- Silva, A. M., Caminhas, W., Lemos, A., & Gomide, F. (2014). A fast learning algorithm for evolving neo-fuzzy neuron. *Applied Soft Computing*, 14, 194-209.
- Singhal, A., Seborg, D. E. (2002). Pattern matching in multivariate time series databases using a moving-window approach. *Industrial & Engineering Chemistry Research*, 41(16), 3822-3838.
- Soares, A. S., Galvão, R. K. H. (2010). Fault detection using linear discriminant analysis with selection of process variables and time lags. *International Conference on Industrial Technology, ICIT2010*, 217-222.
- Tsymbol, A., Cunningham, P., Pechenizkiy, M., Puuronen, S. (2003). Search strategies for ensemble feature selection in medical diagnostics. *Proceedings of the 16th Symposium on Computer-Based Medical Systems, CBMS2003*, 124-129.
- Vale, K. M. O., Neto, A. F., Canuto, A. M. P. (2010). Using a reinforcement-based feature selection method in classifier ensemble. *Proceedings of the 10th International Conference on Hybrid Intelligent Systems, HIS2010*, 213-218.
- Yang, S. K., Liu, T. S. (1998). A Petri net approach to early failure detection and isolation for preventive maintenance. *Quality and Reliability Engineering International*, 14(5), 319-330.
- Zadeh, L. (1994). Fuzzy logic, neural networks, and soft computing. *Fuzzy systems*, 37(3), 77-84.

## CHAPTER VI

### Chapter 6: Conclusions and Suggested Future Works

#### Abstract

This chapter presents the final conclusions and some important considerations regarding this thesis work. Initially, in section 6.1, there are presented arguments that show the fulfillment of the objectives defined in the beginning of this works. Section 2.1 shows the main contributions resulting from the research carried out and presented in previous chapters. Finally, in section 6.3, some suggestions of future works are presented for further demonstrations of the ability of RNSP in adapting for faults detection, diagnosis and correction problems.

**Keywords:** Goals Achieved; Contributions; Future Works.

## CAPÍTULO VI

### Capítulo 6: Conclusões e Sugestões para Trabalhos Futuros

#### Resumo

Neste capítulo são apresentadas as conclusões finais e algumas considerações importantes sobre o trabalho de tese realizado. Inicialmente, na seção 6.1, apresentam-se argumentos que mostram o cumprimento dos objetivos definidos no início dos trabalhos. Na seção 2.1 são evidenciadas as principais contribuições resultantes da pesquisa realizada e apresentadas nos capítulos anteriores. Finalmente, na seção 6.3, algumas sugestões de trabalhos futuros são apresentadas para demonstrações adicionais da capacidade das RNSP se adaptarem para os problemas de detecção, diagnóstico e correção de falhas.

**Palavras-chave:** Objetivos Alcançados; Contribuições; Trabalhos Futuros.

## 6.1 Objetivos Alcançados

Esta tese tem como eixo principal de trabalho a proposição de sistemas de detecção e diagnóstico de falhas em sistemas dinâmicos multivariáveis. A abordagem usada foca nos modelos neurais sem pesos como classificadores de padrões representativos do estado de operação normal e das condições de falhas considerados em estudos de casos univariável e multivariáveis.

Historicamente os estudos realizados com os modelos neurais sem pesos voltaram seus objetivos para o reconhecimento e classificação de padrões, majoritariamente para o reconhecimento de imagens digitais em ambientes estáticos. Somente nos últimos anos é que surgiram alguns trabalhos envolvendo reconhecimento em ambientes dinâmicos, a exemplo de panos de fundos por câmeras de vídeos e agrupamento de fluxos de dados históricos.

Incentivado pela ausência de trabalhos voltados para a detecção e diagnóstico de falhas com o uso das RNSP, embora exista uma variedade significativa de trabalhos para o problema de FDD com uso dos modelos neurais com pesos; e pelo bom desempenho dessas redes para o reconhecimento e classificação de padrões, surgiu a motivação para o desenvolvimento desta pesquisa. Com esse propósito, a pesquisa realizada tem como resultados a proposição de três sistemas de FDD em ambientes dinâmicos univariável e multivariáveis, com base nas redes neurais sem pesos, alcançando, assim, os objetivos pretendidos no início dos trabalhos. Os três sistemas de FDD propostos foram apresentados com detalhes nos capítulos 3, 4 e 5 desta tese, respectivamente.

## 6.2 Contribuições

A principal contribuição com o desenvolvimento deste trabalho refere-se à viabilidade de adaptação das RNSP para os problemas de detecção e diagnóstico de falhas em sistemas dinâmicos multivariáveis. Além disso, outros modelos computacionais importantes são usados no pré-processamento dos dados e contribuem para a melhoria da acurácia nos resultados obtidos com os sistemas de FDD propostos nos capítulos 3, 4 e 5 desta tese. Com isso, outras contribuições importantes observadas com este trabalho são:

- a) Uma boa diferenciação dos padrões de entrada em sistemas dinâmicos multivariáveis pode ser uma tarefa difícil e, muitas vezes, complexa. Nos sistemas de FDD aqui propostos, o uso da seleção de atributos como etapa prévia, principalmente em uma seleção de atributos por classe, resulta numa importante contribuição para a melhoria dos resultados. O algoritmo RecPun com as modificações propostas seleciona os atributos de maior relevância para cada classe considerada na aplicação. Essa diferenciação nos atributos por classe eleva a qualidade da detecção e classificação realizadas pelos discriminadores das RNSP testadas em cada um dos três sistemas de FDD proposto;
- b) Os modelos de pré-processamento dos atributos selecionados presentes no sistema de FDD-WiSARD, apresentado no capítulo 3, também são importantes para a composição dos padrões comportamentais postos como entradas para as RNSP. Esses modelos de pré-processamento utilizam estatísticas obtidas com o processamento de séries temporais formadas a partir dos fluxos de dados relacionados às variáveis (atributos) selecionadas previamente. No sistema FDD-RNSPG, apresentado no capítulo 4, os atributos selecionados passam por um processo de granulação com base na matemática intervalar e na lógica *fuzzy*. O algoritmo usado para a granulação dos atributos selecionados representa também numa importante contribuição para a formação dos



padrões comportamentais apresentados à RNSP e, conseqüentemente, na diferenciação pelos discriminadores das RNSP usadas;

- c) No sistema FDD-pRAM, apresentado no capítulo 5, os atributos selecionados passam por um processo de granulação com base nos conceitos de *clouds* e da lógica *fuzzy*. O sistema de granulação usa recorrência e densidades locais e global para a formação dos padrões comportamentais apresentados à RNSP. O algoritmo recursivo é uma importante contribuição para a diferenciação dos padrões comportamentais pelos discriminadores das redes sem pesos utilizadas;
- d) No contexto das RNSP, o dispositivo WiSARD representa uma ferramenta poderosa no reconhecimento e classificação de padrões e, com este trabalho, para a detecção e diagnóstico de falhas realizados. Os contadores armazenados como conteúdos nos nodos padrão RAM ou as probabilidades nos nodos MPLN ou pRAM adaptam-se bem às características dinâmicas existentes nos processos monitorados. Como mencionado no final da seção 5.1, as redes pRAM com entradas contínuas e aprendizado por reforço apresentam habilidades de aproximadores universais de funções contínuas. A rede WiSARD com nodos contador, MPLN ou pRAM representa uma importante contribuição para os problemas de detecção e diagnóstico de falhas;
- e) A técnica de *bleaching* usadas nos três sistemas de FDD aqui propostos representa uma valiosa contribuição para o “refinamento” nas saídas das RAM e ameniza significativamente os efeitos do uso do critério de aleatoriedade para o desempate entre os discriminadores representantes das classes consideradas. Transpondo essas habilidades para a definição dos limites de classes em problemas de classificação ou agrupamento de padrões, percebe-se a importância do uso da técnica de *bleaching* para elevar a acurácia nas detecções e diagnósticos de falhas, realizados nesta tese.

### 6.3 Sugestões para Trabalhos Futuros

Esta tese investigou problemas de detecção e diagnóstico de falhas em sistemas dinâmicos baseados em redes neurais sem pesos com aplicações univariável e multivariáveis. Além disso, a abordagem proposta contou com a contribuição da seleção de atributos e da granulação de dados como etapas prévias para a formação dos padrões comportamentais apresentados às RNSP. No entanto, como todo trabalho de pesquisa, existem lacunas que possibilitam investigações futuras e a continuidade das pesquisas relacionadas ao tema estudado. Assim, algumas sugestões para trabalhos futuros são postas a seguir:

- a) Investigar a viabilidade de outros modelos de neurônios e arquiteturas neurais sem pesos para a proposição de sistemas de FDD para estudos de caso univariável e multivariável;
- b) Avaliar a relação existente entre os modos de armazenamento dos conteúdos nas memórias RAM com as características dinâmicas existentes nos problemas de detecção e diagnóstico de falhas;
- c) Avaliar o desempenho das RNSP para problemas de FDD sem as etapas prévias de seleção de atributos e/ ou granulação de dados;
- d) Testar as abordagens propostas com outras bases de dados obtidas para configurações diferenciadas nos estudos de casos considerados, principalmente para o TEP e o CSTR;
- e) Usar a rede autoWiSARD ou outro modelo neural sem pesos auto-organizável com aprendizado não supervisionado para a detecção e diagnóstico de falhas e comparar com os resultados aqui obtidos;

- f) Investigar as habilidades das redes neurais sem pesos para a detecção de ponto de mudança em dados históricos com aplicações na detecção de falhas incipientes;
- g) Testar a capacidade dos sistemas de FDD aqui propostos em detectar falhas diferentes das usadas na fase de treinamento e falhas concorrentes;
- h) Abordar a detecção e o diagnóstico de falhas com RNSP no contexto dos sistemas de comitê de classificadores (*ensembles*);
- i) Avaliar os sistemas FDD aqui propostos no contexto dos sistemas híbridos e verificar a possibilidade de usar as RNSP com outras técnicas como algoritmos genéticos, redes bayesianas, sistemas especialistas, dentre outras;
- j) Aplicar a metodologia completa aqui proposta a outros estudos de caso reais, confirmando a sua generalidade e importância para a realização de projetos práticos reais, e aprimorando-a quando necessário.

## Abreviaturas

CSTR:	<i>Continuous Stirred Tank Reactor.</i>
FDD:	<i>Fault Detection and Diagnosis.</i>
FDD-pRAM:	Detecção e Diagnóstico de Falhas com RAM Probabilísticas.
FDD-RNSPG:	Detecção e Diagnóstico de Falhas com Redes Neurais Sem Pesos Granulares.
FDD-WiSARD:	Detecção e Diagnóstico de Falhas com Base no Dispositivo WiSARD.
MPLN:	<i>Multiple-valued Probabilistic Logic Neuron.</i>
pRAM:	<i>probabilistic RAM.</i>
pRAM:	<i>probabilistic RAM.</i>
RAM:	<i>Random Access Memory.</i>
RecPun:	RECompensa/ PUNição.
RNSP:	Rede Neural Sem Pesos.
TEP:	<i>Tennessee Eastman Process.</i>
WiSARD:	<b><i>Wilkes-Stonham-Aleksander Recognition Device.</i></b>

**UFBA**  
**UNIVERSIDADE FEDERAL DA BAHIA**  
**ESCOLA POLITÉCNICA**

**PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA INDUSTRIAL - PEI**

Rua Aristides Novis, 02, 6º andar, Federação, Salvador BA

CEP: 40.210-630

Telefone: (71) 3283-9800

E-mail: [pei@ufba.br](mailto:pei@ufba.br)

Home page: <http://www.pei.ufba.br>

